Das KOMA-Script Paket

Frank Neukam Markus Kohm

1997/11/25

Inhaltsverzeichnis

1 Allgemeines **4**
- 1.1 Einleitung . . . 4
- 1.2 Danksagung . . . 5
- 1.3 Rechtliches . . . 5
- 1.4 Kompatibilität zu den Originalclasses . . . 6

2 Das neue Layout **7**
- 2.1 Die neuen Fonts . . . 7
- 2.2 Der Satzspiegel . . . 7
  - 2.2.1 Grundlegendes . . . 7
  - 2.2.2 Konstruktion . . . 8
  - 2.2.3 Auswahl des gewünschten Satzspiegels . . . 9
  - 2.2.4 Ungewöhnliche Satzspiegel . . . 12
  - 2.2.5 Berücksichtigung von Kopf- und/oder Fußzeile . . . 12
  - 2.2.6 Andere Papierformate . . . 13
  - 2.2.7 Einbindung der Befehle zur Satzspiegeleinstellung . . . 15
- 2.3 Sonstige Änderungen . . . 16
  - 2.3.1 Default-Fontgröße . . . 16
  - 2.3.2 Kapitelüberschriften und Numerierung . . . 16
  - 2.3.3 Kolumnentitel und Pagina . . . 17
  - 2.3.4 Kleinere kosmetische Veränderungen . . . 18

3 Zusätzliche Befehle **21**
- 3.1 Die Erweiterungen des Abstracts und der Titelseite . . . 21
- 3.2 Gestaltung des Inhaltsverzeichnisses . . . 22
- 3.3 Die „xxxxfont“-Variablen . . . 24
- 3.4 Die „marginline“ . . . 24
- 3.5 Der „addchap“- bzw. „addsec“-Befehl . . . 25
- 3.6 Der „minisec“-Befehl . . . 25
- 3.7 Das „labeling“-Environment . . . 26

4 Erweiterung der Internationalisierung **27**

5 Die scrlettr-class **28**
5.1 Das Original 28
5.2 Zusätzliche Eigenschaften der scrlettr-class 29
5.2.1 Mehrsprachigkeit 29
5.2.2 Zweiseitiger Druck 32
5.2.3 Layout 32
5.2.4 Adreßfeld und Zusatz-„location“-Feld 33
5.2.5 Referenzfeld 35
5.2.6 Titel und Betreff 35
5.2.7 Seitenstile 36
5.2.8 Adreßdateien 36
5.2.9 Serienbriefe 39
5.2.10 Das scraddr-package zur Auswertung von Adreßdateien 40
5.2.11 Adreßdatenbankkonverter 41
5.3 Befehls- und Variablenübersicht 44

6 Das scrpage-package **48**
6.1 Die Anwenderschnittstelle 48
6.2 Die Expertenschnittstelle 50
6.3 Eigene Seitenstile und Kapitelanfangsseiten 52

7 Die kleineren Beigaben **53**
7.1 Das scrdate-package 53
7.2 Das scrtime-package 54

Literaturverzeichnis **55**

Index **56**

1 Allgemeines

1.1 Einleitung

Ursprünglich erstellt wurde das SCRIPT-Paket von Frank Neukam für LaTeX 2.09. Ziel war damals ein Style für die Erstellung eines Vorlesungs-Skripts.

Zur selben Zeit bekam er Jan Tschicholds Buch über Typographie [JTsch87] in die Hand. Damit stand für Frank fest, daß es an der Zeit war, endlich ein paar Styles zu erstellen, die auch die europäischen Regeln der Typographie und Papierformate berücksichtigen.

Version 1.0 seiner Document-Style-Famile SCRIPT veröffentlichte er am 6. Juni 1992 per ftp. Die Resonanz darauf war enorm.

Irgendwann zwischen Ende 1992 und Anfang 1993 erreichte diese Familie auch meinen Rechner. Nachdem ich bereits vorher mit verschiedenen Document- und Option-Styles experimentiert hatte, stand für mich sofort fest, daß diese Familie ab sofort als Grundlage für alle meine weiteren Tätigkeiten dienen sollte.

Da ich allerdings im Gegensatz zu Frank Neukam nicht das Format DIN A4 bzw. ISO A4 sondern A5 bevorzugte, dauerte es nicht lange, bis ich seine Familie um eine variable Seitengrößenoption erweitert hatte.

Es folgten andere Erweiterungen und Ergänzungen. Ich selbst erstellte auch einen Document-Style für eine Literaturgruppe.

Im Dezember 1993, als die Versionsvielfalt um sein Paket gerade in ein Chaos auszubrechen drohte, brachte Frank Neukam die Version 2.0 heraus.

Wie jeder weiß, folgte kurz darauf in Punkto LaTeX ein Quantensprung. Die Rufe nach einer neuen, angepaßten Version wurden laut. Kurz darauf war der erste schnelle Hack in Umlauf, der SCRIPT 2.0 als SCRIPT 2.0E für LaTeX 2_ε verfügbar machte.

Nun bin ich ein ewiger — wenn auch sehr chaotischer — Perfektionist, der mit solchen Notlösungen nie ganz zufrieden ist. Ich wollte eine Lösung, die sowohl möglichst konform zu SCRIPT 2.0 als auch zu LaTeX 2_ε ist. Also machte ich mich an die Arbeit.

Die erste Version des Pakets war am 7. Juli 1994 fertig. Diese Version schickte ich unverzüglich an Frank Neukam, um sein OK oder Änderungsvorschläge zu bekommen. Unter allen Umständen wollte ich ein neues Versionschaos vermeiden. Inzwischen hatte ich auch schonmal Kontakt zu Frank, der KOMA-SCRIPT dabei abgesegnet hat.

Jedenfalls sah ich mich gezwungen, meine Version unter anderem Namen und — aufgrund der vielen Nachfragen — möglichst rasch zu veröffentlichen. Daraufhin jagte

eine Korrektur die nächste.

Von Anfang an hatte ich den Ehrgeiz gehabt, ebenfalls ein vollständiges Paket und wenn irgend möglich mit Hilfe von doc/docstrip zu erstellen. Nachdem Axel Kielhorn Anfang Dezember 1994 die bisher vermißte Letter-Class beigesteuert hatte, machte ich mich also an den letzten Schritt, der hiermit vollendet sein sollte.

Weite Teile dieser Anleitung sind direkt Frank Neukams Anleitung zu SCRIPT 2.0 entnommen.

Bis auf weiteres erkläre ich mich für KOMA-SCRIPT verantwortlich. Inzwischen stecken einige hundert Stunden Arbeit in dem gesamten Paket. Dennoch ist es noch lange nicht perfekt oder auch nur vollkommen fehlerfrei. Alle Fehlermeldungen und Verbesserungs- sowie Erweiterungsvorschläge können entweder an meine e-mail-Adresse, in der news-group de.comp.tex oder an meine Postanschrift gerichtet werden:

MausNet : Markus Kohm @ HD

Gelbe Post: Markus Kohm, Fichtenstraße 63, 68535 Edingen-Neckarhausen

1.2 Danksagung

Ich möchte an dieser Stelle eigentlich nur einen explizit nennen. Ich danke Frank Neukam für seine Document-Style-Familie SCRIPT. Ohne die Mühe, die er sich einmal gemacht hat, gäbe es auch dieses Paket nicht.

Im übrigen danke ich allen, die bei der Entwanzung und Verbesserung mitgeholfen haben. Dank gebührt auch dem LaTeX-Team und Dante; ersterem für die notwendige Erneuerung, letzteren für den CTAN-Server, über den ich das Paket verbreiten darf.

Diese Anleitung wurde von Harald Sommerfeldt überarbeitet und um einige Teile ergänzt, dem ich hierfür ebenfalls herzlich danke.

Werner Lemberg darf ebenfalls nicht vergessen werden. Er hat weite Teile der Anleitung ins Englische übersetzt.

1.3 Rechtliches

Die Weitergabe und Benutzung dieses Paketes ist erlaubt, soweit diese zu den für LaTeX üblichen Bedingungen erfolgt. Dies bedeutet insbesondere, daß sich niemand an diesem Paket bereichern darf! Das Paket darf auch nur in *vollständiger* Form weitergegeben werden. Änderungen an einzelnen Dateien sind erlaubt, wenn diese dokumentiert und die betroffenen Dateien umbenannt werden. Es muß dennoch immer eine unveränderte Version im Paket enthalten sein.

Ich lehne jegliche Haftung für Schäden, die aus der Verwendung des Paketes oder eines Teils daraus resultieren, ab. Die Verwendung geschieht ausschließlich auf eigene Gefahr des Anwenders. Zwar bemühe ich mich immer, möglichst wenig Fehler in meine

Standard-Class	SCRIPT 2.0-Style	KOMA-SCRIPT-class
article	script_s	scrartcl
report	script	scrreprt
book	script	scrbook
letter	script_l	scrlettr

Tabelle 1.1: Zuordnung von Standard-class bzw. SCRIPT 2.0-Style zu KOMA-SCRIPT-class.

Werke einzubauen, aber leider ist Fehlerfreiheit nahezu unmöglich. So kann es immer zu unerwünschten Effekten kommen.

1.4 Kompatibilität zu den Originalclasses

Alle Befehle, die in den zugrundeliegenden Standard-Classes vorhanden sind, sind auch im KOMA-SCRIPT-Paket enthalten. Es wurden allerdings einige weitere Befehle hinzugefügt. Einige Befehle verhalten sich auch etwas anders als bei den Standard-Classes. Dennoch sollte es möglich sein, jedes Dokument, das für die Standard-Classes geschrieben wurde, auch mit den classes aus dem KOMA-SCRIPT-Paket zu bearbeiten.

Gegenüber den Document-Styles aus der SCRIPT 2.0 Familie wurden nur wenige Änderungen im Befehlsumfang vorgenommen. Einige wenige Befehle, die meiner Meinung nach eher Optionen sein sollten, wurden zu Optionen. Im Kompatibilitätsmodus für LaTeX 2.09 sind diese Befehle aber noch vorhanden.

Die Zuordnung zwischen Standard-Class und KOMA-SCRIPT-Class zeigt Tabelle 1.1.

2 Das neue Layout

2.1 Die neuen Fonts

`\sfb`

Überall, wo in den Standardklassen **Extended Boldface** (`\bfseries`) verwendet wurde (also in allen Überschriften, im Inhaltsverzeichnis und im `description` Environment), wird jetzt standardmäßig ein neuer Font benutzt, nämlich **Sans Serif Bold Extended** (`\sffamily\bfseries`). Es handelt sich dabei um eine serifenlose Schrift in halbfetter Ausführung, die durch den Befehl `\sfb` nach dem alten Auswahlverfahren selektiert wird. So wie alle anderen Befehle nach dem alten Auswahlverfahren (`\rm`, `\sl`, `\sf`, `\tt`, `\it`, `\bf`, `\em`) sollte auch `\sfb` eigentlich nicht mehr verwendet werden. Zu beachten ist lediglich, daß bei den CM-Fonts als kleinste Größe ein 10pt-Font zur Verfügung steht; wird beispielsweise mittels `\tiny\sfb` eine kleinere Schrift gewünscht, so wird automatisch die normale Extended Boldface eingestellt.

2.2 Der Satzspiegel

2.2.1 Grundlegendes

Der Satzspiegel definiert die Größe des bedruckten Bereichs einer Seite[1] und dessen Positionierung auf der Seite; seine Breite ist abhängig von der gewählten Schriftart und -größe (optimale Lesbarkeit ist bei ca. 60–70 Zeichen pro Zeile gegeben). Bei seiner Konstruktion ist besonders auf die Seitenproportionen (Verhältnis von Breite zu Höhe) sowie auf seinen Stand auf der Seite zu achten.

Jan Tschichold hat in dem in der Einleitung angegebenen Buch [JTsch87] eine Konstruktion vorgeschlagen, die einfach durchführbar ist und bei jedem Seitenverhältnis des gewählten Papierformats (also auch bei dem „unklassischen“ ISO/DIN A4-Format mit dem Seitenverhältnis $1 : \sqrt{2}$, von dem in allen KOMA-Script-Haupt-classes normalerweise ausgegangen wird) sehr ansprechende Ergebnisse liefert. Dies liegt wohl in der Hauptsache daran, daß hier der Satzspiegel die gleichen Seitenverhältnisse wie die zu bedruckende Seite selbst aufweist; auch die Ränder stehen in einer sehr einfachen Relation

[1] Im folgenden auch „Textbereich“ genannt; Kopfzeile, Fußzeile und Randnoten zählen jedoch nicht dazu.

zueinander (innerer zu äußerer bzw. oberer zu unterer Rand verhalten sich wie 1:2; bei einseitigem Druck, also ohne die `twoside`-Option, sind innerer und äußerer Rand gleich groß).

2.2.2 Konstruktion

Die Konstruktion basiert auf einer Einteilung der Breite und Höhe der Seite in n gleiche Teile. Das Ergebnis ist je eine Längenangabe für die Horizontale — im folgenden mit HLE (horizontale Längeneinheit) abgekürzt — und die Vertikale (VLE).

Beispiel: Für eine DIN A4-Seite (210 mm·297 mm) ergibt sich bei Zehnteilung der Seite ($n = 10$) für die horizontale Längeneinheit HLE = 210 mm/10 = 21 mm und für die vertikale Längeneinheit VLE = 297 mm/10 = 29,7 mm.

Nun wird bei doppelseitigem Druck einfach der innere Rand[2] auf eine Breite von 1 HLE und der äußere Rand auf 2 HLE gesetzt. Soll einseitig gedruckt werden, so wird sowohl der innere als auch der äußere Rand auf 1,5 HLE eingestellt. Analog dazu wird ein oberer Rand von 1 VLE und ein unterer Rand von 2 VLE freigelassen. Für den Textbereich bleibt so eine Fläche von $(n-3)$ HLE Breite mal $(n-3)$ VLE Höhe. Randbemerkungen („marginal notes“) stehen im äußeren Rand und sind auf eine Breite von 1,5 HLE (bei Verwendung der `twoside`-Option) bzw. 1 HLE (sonst) begrenzt.

Beispiel: Bei Sechsteilung der Seite ($n = 6$) sind der linke und rechte Rand zusammen 3 HLE und der Textbereich ebenfalls $6 - 3 = 3$ HLE breit. Das gleiche gilt für die vertikalen Größen. Damit ist bei einer Sechsteilung der Seite der Textbereich genau halb so breit und hoch wie die gesamte Seite.

Die TEX-Forderung, daß die Textbereichsbreite (`\textwidth`) ein Vielfaches der Zeichenbreite der `Typewriter`-Schrift sein soll, wird durch die beschriebene exakte Konstruktion nicht erfüllt. Sie wird allerdings selbst von den Original-Styles nicht eingehalten und ist nur bei komplett in `Typewriter`-Schrift geschriebenen Zeilen von Belang. In Abschnitt 2.2.4 wird eine Möglichkeit aufgezeigt, wie beispielsweise für Programmlistings der Satzspiegel doch noch der `\tt`-Schrift angepaßt werden kann.

Die zweite Regel der Satzspiegelkonstruktion, daß `\textheight` (die Höhe des Textbereiches) gleich `\topskip` vermehrt um ein ganzzahliges Vielfaches des Grundlinienabstandes (`\baselineskip`) sein soll, wurde jedoch befolgt. Damit wird erreicht, daß bei komplett mit Fließtext gefüllten Seiten (also Seiten ohne Überschriften, Tabellen, *displayed material* oder ähnlichem) die Absätze nicht auseinandergezogen werden müssen, um zu gewährleisten, daß die unterste Zeile mit dem unteren Rand des Textbereichs zusammenfällt. Der Satzspiegel kann dadurch maximal eine knappe Zeile höher werden als dies bei einer exakten Konstruktion der Fall gewesen wäre.

[2] Es ist hier der effektiv sichtbare Rand gemeint, nicht die odd- bzw. evensidemargin-Einstellung; diese Größen werden intern so errechnet, daß eben genau der gewünschte Rand sichtbar ist.

2.2.3 Auswahl des gewünschten Satzspiegels

Man erkennt leicht, daß für kleine n auch der Textbereich im Verhältnis zur Seitengröße klein ist, und für sehr große n der Textbereich die Seite nahezu ausfüllt (Randbemerkungen sind ausgehende von ISO A4 bei n größer als 12 mangels Platz kaum noch möglich und schon gar nicht mehr sinnvoll umbrechbar). So läßt sich mit lediglich einer Größe der Satzspiegel in weiten Grenzen verändern, und man hat außerdem noch die Garantie, daß er immer optimal konstruiert ist.

Optional läßt sich noch ein Bindungs-Korrektur-Maß *BCOR* (für binding-correction) angeben, so daß bei der Satzspiegelkonstruktion berücksichtigt wird, daß ein Teil der Seite durch Heftung oder Bindung nicht sichtbar ist oder abgefräst wird. Die Seite wird dazu um *BCOR* schmäler betrachtet, als sie wirklich ist, der Satzspiegel auf dieser Basis berechnet, und anschließend der innere Rand um dieses Stück vergrößert.

Beispiel: Ist durch eine Klebebindung, bei der sich die Seiten nicht mehr bis ganz innen aufschlagen lassen, der sichtbare Bereich der Seite nicht mehr 210 mm breit (ISO/DIN A4), sondern 8 mm weniger, so gibt man für *BCOR* 8 mm an (wie und wo steht weiter unten) und schon steht der Satzspiegel beispielsweise bei einseitigem Druck wieder exakt mittig auf dem sichtbaren Teil der Seite und hat auch die gleichen Proportionen wie dieser (202 : 297 statt 210 : 297).

`\typearea[`*BCOR*`]{`*DIV*`}`

Der Satzspiegel wird durch Angabe des Befehls `\typearea` neu konstruiert. Weil interne Größen wie `\oddsidemargin` u.ä. verändert werden, die nur in der Präambel (dem Teil der Textquelldatei vor `\begin{document}`) modifiziert werden dürfen, ist auch der `\typearea`-Befehl nur in der Präambel erlaubt. Die Optionalangabe `[`*BCOR*`]` stellt die gleichnamige Größe zur Bindungs-Korrektur ein, während `{`*DIV*`}` die Rolle des oben genannten Divisors n übernimmt. `[`*BCOR*`]` bedarf einer Maßeinheit, wobei alle bei TEX erlaubten Einheiten möglich sind (pt, mm, cm, in, ...), `{`*DIV*`}` hingegen muß eine ganze Zahl größer als 3 sein.

Bei Verwendung einer der class scrartcl, scrreprt oder scrbook wird `{`*DIV*`}` nach Tabelle 2.1 gewählt (wobei keine Bindungs-Korrektur vorgenommen wird) und der `\typearea`-Befehl durch den `\documentclass`-Befehl automatisch ausgeführt. Dies kommt daher, daß die classes das package typearea, das diese Berechnungen durchführt, automatisch einliest.

Als Anhaltspunkt für die manuelle `{`*DIV*`}`-Einstellung kann Tabelle 2.2 dienen. Dabei wurde $BCOR = 0\text{pt}$ und das Papierformat `a4paper` sowie zweiseitiger Druck zugrundegelegt; die Abstimmung der Textbereichshöhe auf den (von der gewählten Schriftgröße abhängigen) Grundlinienabstand, die die Textbereichshöhe nochmals um maximal 14,5 pt $\simeq$ 5,1 mm vergrößern kann, wurde nicht berücksichtigt.

Textfontgröße:	10pt	11pt	12pt
DIV:	8	10	12

Tabelle 2.1: {*DIV*}-Voreinstellungen für DIN/ISO-A4

	Textbereich		Ränder	
DIV	Breite [mm]	Höhe [mm]	oben [mm]	innen [mm]
6	105.00	148.50	49.50	35.00
7	120.00	169.71	42.43	30.00
8	131.25	185.63	37.13	26.25
9	140.00	198.00	33.00	23.33
10	147.00	207.90	29.70	21.00
11	152.73	216.00	27.00	19.09
12	157.50	222.75	24.75	17.50
13	161.54	228.46	22.85	16.15
14	165.00	233.36	21.21	15.00
15	168.00	237.60	19.80	14.00

Tabelle 2.2: Satzspiegelmaße in Abhängigkeit von *DIV* bei DIN/ISO A4

Will man einen möglichst guten Wert für {*DIV*} wählen, so kann man die Ermittlung auch direkt KOMA-Script überlassen. Dazu verwendet man einfach den Werte 1 oder 2. Diese Werte ergeben normalerweise keine sinnvolle Aufteilung, da ja bereits drei Teile für den Rand benötigt werden. Stattdessen berechnet KOMA-Script dann einen passenden Wert. Dazu wird davon ausgegangen, daß eine optimale Lesbarkeit des Textes bei ca. 69 Zeichen pro Spalte gegeben ist. Gleichzeitig wird dafür gesorgt, daß eine minimale Randbreite nicht unterschritten wird. Neben den Werten 1 und 2 hat der Wert 0 ebenfalls noch eine Sonderbedeutung. Bei 0 wird der voreingestellte Defaultwert für die Papier- und Schriftgröße verwendet. Existiert ein solcher nicht, so wird wie oben Wert berechnet. Dem Wert 3 kommt ebenfalls noch eine Sonderbedeutung zu. Siehe hierzu die Erläuterungen zu Option `DIV` bzw. `DIVclassic` (Seite 11).

`DIV`*Faktor*
`DIVcalc`
`DIVclassic`
`BCOR`*Korrektur*

Die Satzspiegelparameter *DIV* und *BCOR* können alternativ auch als Option an die classes scrartcl, scrreprt und scrbook bzw. das package typearea übergeben werden; ein

expliziter Aufruf von `\typearea` kann dann entfallen. Für *Faktor* bzw. *Korrektur* ist hierbei jeweils das gewünschte Maß direkt hinter der Option anzugeben.

Ein kleines Beispiel hierzu:

```
\documentclass[DIV14,BCOR0.5cm]{scrreprt}
```

Darüber hinaus ist noch die Option `DIVcalc` möglich, mit Hilfe derer dafür gesorgt wird, daß an Stelle von Defaultwerten für `DIV` berechnete Werte verwendet werden. `DIVcalc` entspricht damit weitestgehend `DIV1` (siehe Seite 9).

Desweiteren bietet die Option `DIVclassic` noch eine weitere interessante Möglichkeit. Für klassische Papierformate wird neben der oben beschriebenen Satzspiegelkonstruktion, die vom verwendeten Zeichensatz bzw. der durchschnittlichen Breite und Höhe desselben abhängt, noch eine weitere angeboten, die allein vom Papierformat abhängig ist. Dabei geht man wie schon zuvor bei dem Verhältnis von Textbereich und Papierformat davon aus, daß die größte Harmonie dann zu finden ist, wenn man dieselben Größen möglichst oft wiederfindet. Deshalb wählt man für die Höhe des Textbereichs die (sichtbare) Breite des Papiers. Die konkrete Konstruktion dieses spätmittelalterlichen Buchseitenkanons kann beispielsweise. bei Jan Tschichold nachgelesen werden.[3]Als Nachteil dieser Konstruktion muß betrachtet werden, daß die Schriftgröße unberücksichtigt bleibt. Damit bleibt der Grundsatz, daß eine Zeilenlänge von durchschnittlich 66 bis 72 Zeichen als angenehm empfunden wird, unbeachtet bzw. der Verantwortung des Autors oder Setzers überlassen. Da das typearea-package neuerdings vor größeren Abweichungen des *DIV*-Wertes bzw. der Zeilenlänge vom Ideal warnt, kann man diese Option aber durchaus einmal ausprobieren.

Wird ein ungünstiger *DIV*-Wert gewählt oder ermittelt, der bei der Zeilenlänge zu einer Abweichung von mehr als 15% vom Idealbereich führt, so gibt typearea seit Version 2.4n eine Warnung aus. Ebenfalls wird vor einem *DIV*-Wert kleiner sechs gewarnt, da dabei der Textbereich in Breite und Höhe jeweils weniger als die Hälfte der Seite belegt. Diese Warnung bedeutet nicht zwangsläufig, daß ein Fehler vorliegt. So wird beispielsweise `scrguide.dtx` selbst diese Warnung erzeugen. Dies liegt jedoch daran, daß es sich dabei um ein Dokument mit grundsätzlich anderem Satzspiegel handelt. Bei normalen Dokumenten sollte man die Warnung ernstnehmen und die dabei gegebenen Ratschläge befolgen.

Vergleicht man jedoch die Ergebnisse von `DIVcalc`, `DIVclassic` und den voreingestellten Werten, so wird man feststellen, daß die in der `log`-Datei ausgegebene Abweichung vom Idealwert bei unterschiedlichem *DIV* gleich – auch gleich Null – sein kann. Daraus muß man schließen, daß nach den gegebenen Kriterien verschiedene Werte als

[3]Tatsächlich lesen offenbar immer mehr Anwender dieses Pakets nach, was bekannte Typographen zum Thema Satzspiegel geschrieben haben. Das führte in letzter Zeit zu Nachfragen, ob die im Paket gewählte Satzspiegelkonstruktion etwas mit der *Kreiskonstruktion* zu tun habe. Mich wiederum veranlaßten die Fragen, eben diese Konstruktion – in leichter Abwandlung, da ein ganzzahliger *DIV*-Wert gefunden werden sollte – ebenfalls anzubieten.

ideal betrachtet werden können. Ursache dafür ist, daß die Angabe der idealen Zeilenbreite nicht allein einen Wert (beispielsweise den Durchschnittwert 69) sondern einen ganzen Bereich (66 bis 72 Zeichen pro Zeile) zuläßt.

2.2.4 Ungewöhnliche Satzspiegel

`\areaset[`*BCOR*`]{`*WIDTH*`}{`*HEIGHT*`}`

Wird ein Satzspiegel gewünscht, der ungewöhnliche Proportionen hat oder zwar wie oben angegeben konstruiert werden soll, jedoch *exakt* die in der vorstehenden Tabelle aufgeführten Maße (ohne die Korrektur auf Vielfache des Grundlinienabstandes) haben soll, so gibt es noch alternativ zu `\typearea` den Befehl `\areaset`, der die Textbereichsbreite auf `{`*WIDTH*`}` und die Textbereichshöhe auf `{`*HEIGHT*`}` setzt (jeweils ohne jede nachträgliche Korrektur) und die Rand- und Randnoten-Einstellungen so vornimmt, daß der Textbereich sinnvoll positioniert wird (ähnlich wie beim `\typearea`-Befehl). Auch die Wirkung der optionalen `[`*BCOR*`]`-Angabe ist wie schon oben beschrieben.

Damit ist es nun sehr einfach möglich, beispielsweise genau quadratische oder sehr schmale hohe Satzspiegel zu verwirklichen. Wenn die automatische Positionierung nicht gefällt, kann diese durch die entsprechenden Befehle natürlich auch noch nachträglich verändert werden. Weiterhin ist es hiermit möglich, die Satzspiegelbreite auf die Breite eines vorgegebenen Textes abzustimmen.

Als Beispiel soll, wie weiter oben angekündigt, der Textbereich der Breite von 30 Zeichen in `Typewriter`-Schrift angepaßt werden. Eine Befehlsfolge, die dieses durchführt, würde etwa lauten:

```
\newlength{\thirtytt}
\settowidth{\thirtytt}{\texttt{12345678901234567890123456789O}}
\areaset{\thirtytt}{20cm} % Textbereichsh"ohe: 20cm
```

Auch dieser Befehl darf selbstverständlich nur in der Präambel auftreten.

2.2.5 Berücksichtigung von Kopf- und/oder Fußzeile

Manchmal ist es sinnvoll, daß bei der Konstruktion des Satzspiegels auch die Kopf- und/oder Fußzeile als zum Textbereich dazugehörig betrachtet wird. Dies ist beispielsweise der Fall, wenn der Kolumnentitel wie in dieser Dokumentation durch eine horizontale Linie vom eigentlichen Text getrennt ist und dadurch optisch näher an diesen heranrückt.

`headinclude`
`headexclude`

Bei Angabe der package-Option `headinclude` gilt bei einer nachfolgenden Satzspiegelberechnung mit Hilfe des `\typearea`- oder `\areaset`-Befehls alles, was oben über den Textbereich gesagt wurde, für den Textbereich plus Kopfzeile. Natürlich existiert auch eine gegenteilige Option `headexclude`.

`footinclude`
`footexclude`

Für Fußzeilen gibt es die analogen Optionen `footinclude` und `footexclude`.

Die Optionen können entweder direkt bei `\documentclass` angegeben werden, oder auch bei `\usepackage`, wenn das typearea-package explizit eingeladen wird.

Dadurch, daß bei KOMA-Script dieses Verhalten im Gegensatz zu Script 2.0 nicht mehr mit zusätzlichen Präambel-Befehlen `\headincludeon` und `\headincludeoff` bzw. `\footincludeon` und `\footincludeoff`, sondern mit Optionen geschaltet wird, wird kein expliziter `\typearea`-Befehl mehr benötigt. Seit Version 2.2a von KOMA-Script (Versiondatum [1995/01/25]) können auch `DIV` und `BCOR` per Option angegeben werden. Damit ist der explizite `\typearea`-Befehl kaum noch notwendig.

2.2.6 Andere Papierformate

Für die Einstellung des Papierformats werden bei KOMA-Script nicht mehr die bei Script 2.0 üblichen Option `DINA3`, `DINA4`, `DINA5`, `DINA6` und `DINA3l`, `DINA4l`, `DINA5l`, `DINA6l`, sondern die bei den Standard-classes üblichen Optionen `a4paper`, `a5paper`, `letterpaper`, `landscape`, ... verwendet. Im Gegensatz zu den gleichnamigen Optionen der Standard-classes werden allerdings keine starren Satzseitenspiegel daraus berechnet, sondern mit Hilfe des `typearea`-packages Satzseitenspiegel nach den oben genannten Regeln.

`a0paper`
`b0paper`
`c0paper`
`d0paper`
`\isopaper[`*Basis*`]{`*Halbierwert*`}`

Als Besonderheit werden sind im typearea package nicht nur die die DIN/ISO-Formate A4 und A5 definiert, sondern es ist möglich alle A-, B-, C- und D-Formate ab den 0-Größen (A0, B0, C0 und D0) zu wählen. Dies ist sowohl per Option beim Laden von typearea per `\usepackage` möglich, als auch beim Laden einer der KOMA-Script-Klassen scrbook, scrreprt oder scrartcl. Die Optionen heißen wie gewohnt `a0paper`, `a1paper` usw.

bzw. `b0paper`, `c0paper`, `d0paper` usw. Darüberhinaus kann auch an anderer Stelle (der Präambel) das Papierformat per `\isopaper`-Befehl definiert werden. Für das optionale Argument *Basis* ist jeweils der Kennbuchstaben des Formats als Kleinbuchstabe anzugeben.

`a`: DIN/ISO-A-Format (A0 = 841 mm · 1189 mm)

`b`: DIN/ISO-B-Format (B0 = 1000 mm · 1414 mm)

`c`: DIN/ISO-C-Format (C0 = 917 mm · 1297 mm)

`d`: DIN/ISO-D-Format (D0 = 771mm · 1090 mm)

Der Parameter *Halbierwert* gibt demgegenüber den „Ableitungsfaktor“ an. Für DIN/ISO-C6 wäre also beispielsweise einfach `\isopaper[c]{6}` zu schreiben. Bei Verzicht auf den optionalen Parameter wird das A-Format gewählt.

Sollen nun andere, nicht im typearea package enthaltene Papierformate verwendet werden, so muß dafür auch kein eigenes package mehr erstellt werden. Für kleine, einmalige Änderungen kann das Papierformat explizit über die Längenvariablen `\paperwidth` und `\paperheight` angegeben werden. In diesem Fall muß der Satzseitenspiegel anschließend mit dem `\typearea`-Befehl oder dem `\areaset`-Befehl berechnet werden. Es ist aber auch möglich weitere Optionen in eine Datei `typearea.cfg` zu schreiben. Eine solche Option kann dann wie die Standardoptionen verwendet werden.

`\SetDIVList{`*list*`}`

Außer der Seitengröße kann für jede Option oder auch generell ein von der Schriftgröße abhängiger *DIV*-Faktor definiert werden. Dazu definiert man mit dem Befehl `\SetDIVList` eine entsprechende Liste. Die Liste besteht aus einer Reihe von Werten in geschweiften Klammern, die einfach aneinandergereiht wird. Der Wert ganz links ist für 10pt-, der nächste Wert für 11 pt-, der dritte für 12 pt-Schrift usw. Wird keine Liste mit `\SetDIVList` gesetzt, so entspricht dies `\SetDIVList{{8}{10}{12}}`. Ist für eine Schriftgröße kein Standardwert gesetzt, so wird statt dessen 10 genommen.

Beispiel 1: Ein Gedichtbändchen soll im ISO/DIN A5-Format gedruckt werden. Dazu genügt nun die Angabe von `a5paper` als Option, beispielsweise:

```
\documentclass[a5paper]{scrreprt}
```

Beispiel 2: Der Textbereich eines Zeitplanes im ISO/DIN A4 *Quer*format soll exakt 270 mm mal 190 mm groß sein. Dies wird erreicht durch die Angabe von:

```
\documentclass[a4paper]{scrreprt}
\areaset{270mm}{190mm}
```

und einer (systemabhängigen) Anweisung an den Druckertreiber, im Querformat zu drucken.

Beispiel 3: Beim Gedichtbändchen aus Beispiel 1 soll zusätzlich eine Abfräsung von 1,5 mm berücksichtigt werden. Dazu muß der Satzspiegel nochmal neu konstruiert werden, beispielsweise mit:

```
\documentclass[a5paper,DIV12,BCOR1.5mm]{scrreprt}
```

Beispiel 4: Es soll auf Endlospapier mit den Maßen $8\frac{1}{4}$ inch mal 12 inch gedruckt werden, und auch dort soll der Satzspiegel ordentlich aussehen. Mit den Befehlen:

```
\paperwidth 8.25in
\paperheight 12in
\typearea{10}
```

wird dies erreicht. Druckt man häufiger auf Endlospapier, so empfiehlt es sich, diese Anweisungen in eine eigene Option in `typearea.cfg` zu schreiben, die man dann einfach beispielsweise mit:

```
\documentclass[endlos]{scrreprt}
```

verwenden kann. Soll zusätzlich das Argument *DIV* des `\typearea`-Befehls abhängig von der gewählten Schriftgröße sein, so fügt man zusätzlich noch einen `\SetDIVList`-Befehl zu. Der Eintrag in `typearea.cfg` könnte dann so aussehen:

```
\DeclareOption{endlos}
  {\setlength\paperheight {8.25in}%
   \setlength\paperwidth  {12in}%
 \SetDIVList{{8}{10}{12}}}
```

2.2.7 Einbindung der Befehle zur Satzspiegeleinstellung

Im Gegensatz zu den Document-Styles aus der SCRIPT 2.0-Familie ist die Satzseitenspiegelberechnung in den classes des KOMA-SCRIPT-Pakets nicht mehr direkt enthalten. Statt dessen wird von den classes einfach das typearea-package angefordert, in dem diese Berechnungen stattfinden. Die Schriftgröße wird aus den entsprechenden Optionendateien der Standard-classes gelesen. Da die letter-class scrlettr einen besonderen und vor allem festen Satzspiegel besitzt, wurde bei ihm auf die Einbindung des `typearea`-packages verzichtet. Es sollte dann auch *keinesfalls* mit `\usepackage{typearea}` nachgeladen werden.

Hingegen sollte es problemlos sein, das `typearea`-package auch zusammen mit anderen classes, beispielsweise den Standard-classes, zu verwenden, die von keinem festen Layout ausgehen. Damit wird die Satzseitenspiegelberechnung auch für jene classes verfügbar.

Im Kompatibilitätsmodus sind die Befehle `\headincludeon` und `\headincludeoff` bzw. `\footincludeon` und `\footincludeoff` weiterhin vorhanden.

2.3 Sonstige Änderungen

2.3.1 Default-Fontgröße

Wird in der Definition der Document-class nichts anderes angegeben, so wird standardmäßig als Textfont der 11 pt-Font verwendet (wie beispielsweise. in dieser Dokumentation). Eine Ausnahme hiervon bildet wieder die scrlettr-class, bei der ohne Optionsangabe die 12 pt-Schrift benutzt wird.

Bei den Standard-classes werden hingegen normalerweise 10 pt-Fonts verwendet.

2.3.2 Kapitelüberschriften und Numerierung

Kapitelüberschriften in den classes scrbook und scrreprt werden ohne die bei book und report üblichen eigenen Zeilen für das Wort „Kapitel“ und die Kapitelnummer gesetzt, sondern wie bei Überschriften niedrigerer Priorität mit der Kapitelnummer direkt vor der eigentlichen Überschrift. Da hierdurch ein Anhang leicht für ein normales Kapitel gehalten werden kann (er unterscheidet sich von diesen ja nur durch die Art der Numerierung), ist es empfehlenswert, ihn mit `\part*{Anhang}` deutlich vom Rest des Dokumentes abzusetzen.

`bigheadings`
`normalheadings`
`smallheadings`

Die Überschriften werden sowohl bei den Standard-Classes als auch bei KOMA-Script normalerweise recht groß gesetzt. Es gibt offenbar viele Leute, denen das nicht gefällt. Insbesondere bei kleinen Papiergrößen sind die großen Überschriften eher störend. Deshalb wurden die drei Optionen `bigheadings`, `normalheadings` und `smallheadings` implementiert. Über diese ist es möglich drei verschiedene Größenabstufungen der Überschriften einzustellen. Voreingestellt ist `bigheadings`, wobei auch hierbei die Kapitelüberschrift (nur bei scrbook und scrreprt) zusätzlich um eine Stufe gegenüber Version 2.2g (Versionsdatum [1996/01/14]) verringert wurde (siehe auch Abschnitt 3.3 auf Seite 24).

`\raggedsection`

Seit Version 2.2h werden alle Überschriften ohne Trennung gesetzt. Verantwortlich ist dafür der Befehl `\raggedsection`. Will man Trennungen in Überschriften erlauben, so kann man dies beispielsweise mit dem Befehl

```
\let\raggedsection\relax
```

erreichen. Normalerweise ist `\raggedsection` mit `\raggedright` identisch.

`pointednumbers`
`pointlessnumbers`

Im Gegensatz zu früheren Versionen von KOMA-Script und auch Script 2.0 steht nach einfachen Abschnittsnummern ab der Version 2.2c (Versionsdatum [1995/05/25]) nun kein Punkt mehr (einheitlich: 1, 1.1, 1.2.3 usw., siehe [Duden20, Seite 18, Regel 5]), wenn ausschließlich mit arabischen Ziffern numeriert wird. Werden bei der Numerierung auch römische Zahlen bzw. Buchstaben verwendet, so steht nach allen Abschnittsnummern ein Punkt (einheitlich: I., I.1., A., A.1.1. usw. aber auch 1., 1.1., 1.2.3. usw., siehe [Duden20, Seite 18, Regel 6]). Dies gilt selbstverständlich auch für Einträge in Kolumnentitel und Inhaltsverzeichnis.

Über die Optionen `pointednumbers` und `pointlessnumbers` kann jedoch die Numerierung nach Duden abgeschaltet werden. Statt dessen wird dann bei `pointednumbers` immer und bei `pointlessnumbers` nie ein abschließender Punkt gesetzt. Die Optionen heben sich *nicht* gegenseitig auf!

Zu beachten ist noch, daß die korrekte Verwendung des Punktes beim ersten LaTeX-Lauf noch nicht sichergestellt werden kann.

2.3.3 Kolumnentitel und Pagina

Die scrbook-class benutzt als Default den Pagestyle `headings`, der in den Kopfteil der Seiten das aktuelle Kapitel (linke Seiten) bzw. den aktuellen Abschnitt (rechte Seiten) schreibt (sog. lebender Kolumnentitel bzw. running head). Weiterhin wird eine Seitennummer in den Fußbereich der Seite gesetzt (sogenannte Pagina). scrreprt und scrartcl stellen standardmäßig `plain` als Pagestyle ein, jedoch sind natürlich auch hier `headings`, `myheadings` und `empty` verfügbar. Beim Brief stehen teilweise andere Seitenstile zur Auswahl, die in Abschnitt 5.2.7 behandelt werden.

Normalerweise werden Kolumnentitel und Pagina zentriert; wurde jedoch die Option `twoside` angegeben, so wird beides an den äußeren Rand der Seite gesetzt. Bei scrbook ist `twoside` standardmäßig gesetzt.

Die Definitionen der Pagestyles `headings`, `myheadings` und `plain`, sowie die Befehle zum automatischen Setzen der Kolumnentitel durch Befehle wie `\section` oder

`\tableofcontents` wurden dahingehend verändert, daß die Kolumnentitel mit normaler Groß-Klein-Schreibung (vorher alles groß), standardmäßig in *slanted* (vorher *Italics*) und bei Kapiteln ohne das Wort „CHAPTER“ bzw. „KAPITEL“ gesetzt werden. Die Seitennummer wurde generell aus der Kopfzeile in die sonst leere Fußzeile verbannt.

Diese Maßnahmen haben neben der besseren Lesbarkeit („Inhaltsverzeichnis“ liest sich viel angenehmer als „INHALTSVERZEICHNIS“) vor allem den Vorteil, daß nun auch wesentlich längere Abschnitts-Überschriften in die Kopfzeile passen ohne abgeschnitten zu werden.

`headsepline`
`headnosepline`
`footsepline`
`footnosepline`

Wird unter den Kolumnentiteln eine horizontale Linie gewünscht, so ist auch dies jetzt ohne Probleme möglich: sie läßt sich einfach mit der class-Option `headsepline` an- bzw. durch `headnosepline` ausschalten. Das funktioniert natürlich nur bei den Pagestyles `headings` und `myheadings`, da ja alle anderen keinen Kolumnentitel generieren. Durch diese Optionen wird auch `headinclude` automatisch sinnvoll miteingestellt, das heißt `headinclude` bei `headsepline` und `headexclude` bei `headnosepline`. Default ist bei scrbook, scrreprt und scrartcl (bei scrlettr ist wieder alles anders; siehe Abschnitt 5.2.7) `headnosepline`.

Analog gibt es auch die Optionen `footinclude`, `footexclude` für die Fußzeilen, wenn man mit den Optionen `footsepline` bzw. `footnosepline` eine Trennlinie über der Fußzeile ein- bzw. ausschaltet.

Die alten Befehle `\headseplineon` und `\headseplineoff` bzw. `\footseplineon` und `\footseplineoff` sind im Kompatibilitätsmodus weiterhin verfügbar. Da auch im Seitenstil `plain` eine Seitennummer in die Fußzeile geschrieben wird, erscheint die Trennlinie auch dort.

Beispiel: In der vorliegenden Dokumentation wurde mit

```
\documentclass[headsepline]{...}
```

der horizontale Strich unter den Kolumnentiteln eingeschaltet (dadurch wurde auch gleichzeitig `headinclude` ausgeführt).

2.3.4 Kleinere kosmetische Veränderungen

Bis zu Version 2.4l war in Fußnoten ein kleiner horizontaler Bereich (1 em) nur für die Fußnotennummer reserviert, der eigentliche Text sollte also immer erst diesen Abstand

vom linken Rand entfernt stehen. Bei mehrzeiligen Fußnoten sollten die Folgezeilen nochmals 0,5 em links eingezogen werden. Dadurch setzen sich diese besser voneinander ab. Absätze sollten ohne vertikalen Zwischenraum und mit einem Einzug von 1 em (entspricht 1 `\quad`) gesetzt werden. Leider hat das nie so richtig funktioniert, auch wenn das kaum aufgefallen ist. Lediglich in Titeln, die keine eigenen Titelseiten beanspruchten (Voreinstellung in scrartcl), wurde deutlich, daß es keineswegs so funktioniert, wie es gedacht war.

`\deffootnote[`*Markenbereich*`]{`*Einzug*`}{`*Absatzeinzug*`}{`*Fußnotenmarke*`}`
`\deffootnotemark{`*Fußnotenmarke*`}`
`\thefootnotemark`
`\textsuperscript{`*Text*`}`

Als ich dabei war, die Fußnotenausgabe neu zu implementieren, fiel mir auf, daß eigentlich viele Parameter darin leicht veränderbar erreicht werden könnten. Seit Version 2.4l gibt es deshalb den Befehl `\deffootnote`, mit dem man die Gestalt der Fußnoten variieren kann. Der optionale Parameter *Markenbereich* gibt dabei die Größe des linken Einzugs der ersten Zeile des Fußnotentextes an. Rechtsbündig in diesen Bereich wird die Fußnotenmarke gesetzt. Fehlt der Parameter, so wird er automatisch auf den Wert des Parameters *Einzug* gesetzt. Dieser gibt den linken Einzug des gesamten Fußnotentextes an. Im Falle daß Fußnoten mehrere Absätze enthalten, wird mit dem Parameter *Absatzeinzug* der zusätzliche horizontale Einzug von absätzen angegeben. Der letzte Parameter, *Fußnotenmarke*, definiert schließlich die eigentliche Gestalt der Fußnotenmarke in der Ausgabe der Fußnote. Diese wird noch zusätzlich von einer horizontale Box umschlossen.

In allen drei Hauptklassen sind die Fußnoten wie folgt vordefiniert:

```
\deffootnote[1em]{1.5em}{1em}
  {\textsuperscript{\normalfont\thefootnotemark}}
```

Dabei repräsentiert `\thefootnotemark` die eigentliche Fußnotenmarkierung. Mit dem Makro `\textsuperscript` wird die Fußnotenmarkierung als Argument *Text* hoch gesetzt (z. B. „$^{\text{hoch}}$“), `\normalfont` sorgt für den richtigen Zeichensatz (und die richtige Größe). Damit die Fußnotenmarkierungen im Text dem entsprechen, wird mit dem ebenfalls neuen Makro `\deffootnotemark` diese identisch definiert:

```
\deffootnotemark{\textsuperscript{\normalfont\thefootnotemark}}
```

Will man nun erreichen, daß bei mehrzeiligen Fußnoten kein zusätzlicher Einzug ab der zweiten Zeile erfolgt und die Fußnotenmarkierungen nicht hochgestellt und in normaler Fußnotentextgröße erscheinen, so muß man nur die Fußnotenausgabe leicht umdefinieren.

```
\deffootnote{1em}{1em}{\thefootnotemark\ }
```

Der zusätzliche Leerraum am Ende der Definition sorgt dafür, daß die Fußnotenmarke nicht direkt am Text klebt. Nimmt man eine Umdefinierung vor der Ausgabe der Titelseite vor, so wirkt sie auch auf diese. Dies gilt auch für `\deffootnotemark`, allerdings wird dort bei Titelseiten zusätzlich dafür gesorgt, daß die Markierung sich nicht auf die horizontale Zentrierung auswirkt.

Obwohl bei `german` ab Version 2.4 `\frenchspacing` mit verwaltet wird, wird dies zusätzlich eingeschaltet. Es ist jedoch möglich, daß dies in einer der nächsten Versionen entfernt wird.

Bei der `enumerate`-Umgebung wurde in der zweiten Ebene eine Klammer entfernt, so daß „a)'ßtatt „(a)“ geschrieben wird. Weiterhin wurde in der dritten Ebene des `itemize`-Environments der „∗ durch ein „*riangleright*“ ersetzt (das sieht zumindest für Franks und meinen Geschmack etwas gefälliger aus).

Die Autoren- und Datumsangaben in Titeln erscheinen nicht in `\large`, sondern in `\Large`.

In mehrzeiligen Captions (Unterschriften bzw. Überschriften von `table` bzw. `figure`) werden die zweiten und folgenden Zeilen so weit links eingerückt, daß diese nicht mehr unter „Abbildung ...:“, sondern unter dem eigentlichen Text der ersten Zeile beginnen.

Es wurden noch etliche weitere Kleinigkeiten im Layout verändert, die allerdings dem ungeübten Auge kaum auffallen werden. So ist etwa der Spaltenabstand im zweispaltigen Satz von 10 pt auf die übliche 1 cc vergößert worden (1 cc ≃ 12 pt. Will man alles ganz genau wissen, so kann man ja immer noch `scrclass.dtx` studieren (dazu sollte man allerdings auch die niedrigeren TEX-Befehle kennen bzw. zuerst die `.dtx`-Versionen der Original-classes und -packages durcharbeiten).

3 Zusätzliche Befehle

3.1 Die Erweiterungen des Abstracts und der Titelseite

`abstractoff`
`abstracton`

Bei Abstracts besteht nun die Möglichkeit, die automatisch erzeugte Überschrift (beispielsweise „Zusammenfassung“) mit der Option `abstractoff` aus- bzw. mit `abstracton` anzuschalten. Voreingestellt ist bei scrartcl und scrreprt `abstractoff`. Bei scrbook gibt es kein Abstract.

`\extratitle{`*Schmutztitel*`}`
`\titlehead{`*Text*`}`
`\subject{`*Text*`}`
`\publishers{`*Verlag*`}`
`\uppertitleback{`*Text*`}`
`\lowertitleback{`*Text*`}`
`\dedication{`*Widmung*`}`

Der `\maketitle`-Befehl der SCRIPT-Haupt-classes wurde durch die folgenden optionalen Angaben wesentlich erweitert:

- `\extratitle` erzeugt eine eigene Seite, auf der der sogenannte „Schmutztitel“[1] steht. Dieser wird wie normaler Text, also ohne Zentrierung und in normaler Schriftart und -größe gesetzt. Dadurch ist dessen Gestaltung völlig dem Anwender überlassen. Die Rückseite ist auch im zweiseitigen Druck leer.
- `\titlehead` setzt in den Kopfteil der Haupttitelseite einen zusätzlicher Text wie beispielsweise eine Institutsadresse.
- `\subject` setzt direkt oberhalb des eigentlichen Titels eine Zusatzangabe (z.B. „Diplomarbeit“) in etwas größerer Schrift.

[1] Früher war der Buchblock oftmals nicht durch einen Buchdeckel vor Verschmutzung geschützt; diese Aufgabe übernahm dann die erste Seite des Buches, die meist einen Kurztitel, eben jenen „Schmutztitel“, trug. Auch heute noch wird gerne eine Extraseite vor dem eigentlichen Haupttitel für Verlagsname, Buchreihennummer o.ä. benutzt.

- `\publishers` plaziert den Verlagsnamen in den unteren Teil der Haupttitelseite.
- Im zweiseitigen Druck läßt sich die Rückseite des Haupttitels für weitere Informationen (im Beispiel dieser Dokumentation ist dies der Haftungsausschluß) nutzen. Dazu setzt `\uppertitleback` einen Text an den oberen Rand der Seite und `lowertitleback` einen Text an den unteren Rand.
- `\dedication` erzeugt eine seperate Widmungsseite, wobei *Widmung* zentriert und in etwas größerer Schrift gesetzt wird. Wie bei `\extratitle` ist die Rückseite grundsätzlich leer.

Beispiel:

```
\titlehead{%
  Institut f"ur Nachrichtentechnik \hfill SS1993\\
  Hochschulstr.12\\
  3456 Unistadt}
\subject{Diplomarbeit}
\title{Digitale Raumsimulation mit dem DSP56004}
\author{Ulrich Ungenannt}
\lowertitleback{%
  Diese Diplomarbeit wurde mit dem Textsatzsystem
  \TeX\ erstellt.}
\maketitle
```

Der Titel (Kurz- oder Haupttitel) beginnt normalerweise auf Seite 1. Durch Angabe einer Zusatzoption (beispielsweise mit `\maketitle[-1]`) läßt sich die Startseite jetzt beliebig festlegen.

Da es seit LaTeX 2_ε für article eine Option `titlepage` gibt, die auch in scrartcl implementiert ist, entfällt ein `sctitle.sty`.

3.2 Gestaltung des Inhaltsverzeichnisses

Mit Version 2.3h werden zusätzliche Klassenoptionen eingeführt, mit deren Hilfe es möglich ist, Einfluß auf das Inhaltsverzeichnis zu nehmen. Dabei wurden zum einen häufig geäußerte Wünsche, zum anderen aber auch typographische Aspekte berücksichtigt. So werden Dinge, die ich selbst für weniger sinnvoll halte, weniger gut unterstützt.

Die zusätzlichen Optionen führen dazu, daß einige interne Befehle mit Hilfe der Anweisung `\AtEndOfClass` umdefiniert werden. Die Verwendung der Optionen wirkt sich also auf den Speicherverbrauch aus.

`liststotoc`

Normalerweise erscheinen Tabellen- und Abbildungsverzeichnis nicht im Inhaltsverzeichnis. In den meisten Fällen wird dies auch so gewünscht. Ab und zu mag es jedoch sinnvoll sein, diese beiden Verzeichnisse im Inhaltsverzeichnis explizit aufzuführen. Über die Option `liststotoc` wird deshalb die Erzeugung der Überschriften in `\listoftables` und `\listoffigures`, die über die internen Befehle `\lot@heading` und `\lof@heading` erfolgt, am Ende der Class so umdefiniert, daß ein Eintrag (ohne Nummer) in das Inhaltsverzeichnis vorgenommen wird.

`bibtotoc`
`bibtotocnumbered`

Auch das Literaturverzeichnis wird normalerweise nicht im Inhaltsverzeichnis aufgeführt. Mit der Option `bibtotoc` wird dafür gesorgt, daß es ohne Nummer aufgenommen wird. Es gibt jedoch auch Anwender, die das Literaturverzeichnis als eigenständiges Kapitel betrachten, das nicht nur im Inhalsverzeichnis erscheinen, sondern auch mit einer Nummer versehen werden soll. Dies wird mit der Option `bibtotocnumbered` erreicht. In beiden Fällen wird der interne Befehl `\bib@heading` am Ende der Class entsprechend umdefiniert.

Bemerkung: Einige Pakete zur Verwaltung und Gestaltung von Literaturangaben definieren die Literaturumgebung um. Leider wird dabei in der Regel von den Standard-Classes ausgegangen und keinerlei Rücksicht auf Besonderheiten anderer Klassen genommen. In diesem Fall kann es nicht nur dazu kommen, daß die Kolumnentitel unpassend wirken, die beiden hier erwähnten Optionen gehen dann ebenfalls verloren. Abhilfe können hier nur Sie selbst oder der Autor des jeweiligen Pakets schaffen. KOMA-Script ist in diesem Fall nicht der Schuldige.

`idxtotoc`

Der Index kann auf Wunsch ebenfalls in das Inhaltsverzeichnis aufgenommen werden. Er ist jedoch in keinem Fall ein eigenständiges Kapitel, das numeriert werden sollte. Vielmehr dient der Eintrag im Inhaltsverzeichnis eher dazu, darauf hinzuweisen, daß ein Index existiert. Entsprechend wird über die Option `idxtotoc` der interne Befehl `\idx@heading` am Ende der Class umdefiniert. Hier gilt dieselbe Bemerkung wie oben.

Beispiel: Listenverzeichnisse, Index und Literaturverzeichnis in das Inhaltsverzeichnis aufnehmen. Letzteres jedoch nicht als numeriertes Kapitel:

```
\documentclass[liststotoc,idxtotoc,bibtotoc]{...}
```

Variable	Wirkung auf:	Default
`\capfont`	Caption-Texte	`\normalfont`
`\caplabelfont`	Caption-Label	`\normalfont`
`\descfont`	Item-Texte des `description`-Environments	`\sffamily\bfseries`
`\headfont`	Kolumnentitel	`\slshape`
`\pnumfont`	Paginierung	`\normalfont`
`\sectfont`	Sämtliche Überschriften und Titel	`\sffamily\bfseries`

Tabelle 3.1: Die Befehle zur Fonteinstellung, ihre Bedeutung und Standardeinstellungen

3.3 Die „xxxxfont"-Variablen

```
\capfont
\caplabelfont
\descfont
\headfont
\pnumfont
\sectfont
```

Seit der Version 2.0 des SCRIPT-Paketes sind mehrere Variablen zur globalen Schriftänderung in Kolumnentiteln, Paginierung, Captions, Überschriften und dem `description`-Environment bereitgestellt. Dadurch lassen sich mit einem einzigen Befehl beispielsweise sämtliche Überschriften statt in `\sffamily\bfseries` in `\rmfamily` ausgegeben. Dazu muß nur irgendwo am Dokumentanfang (am besten in der Präambel) die Anweisung `\renewcommand{\sectfont}{\rmfamily}` stehen. Auf die gleiche Art und Weise können auch die anderen Fonteinstellungen aus Tabelle 3.1 geändert werden. Selbstverständlich sind nicht nur Fonteinstellungs-, sondern auch Größenänderungsbefehle möglich (siehe auch Abschnitt 2.3.2 auf Seite 16).

3.4 Die „marginline"

`\marginline{`*Randnotiz*`}`

Analog zum `\marginpar` wurde die `\marginline` eingeführt, die für einzeilige Randnotizen gedacht ist. Bei diesen tritt nämlich im `twoside`-Modus auf linken Seiten das Problem auf, daß sie mit `\marginpar` nicht rechtsbündig gesetzt werden (was sinnvoll wäre), sondern durch den Blocksatz einer nicht vollen Zeile linksbündig erscheinen. Bei Verwendung von `\marginline` werden einzeilige Texte sinnvoll in den äußeren Rand gesetzt, was zum Beispiel zum Hervorheben von Querverweisen nützlich ist.

3.5 Der „addchap“- bzw. „addsec“-Befehl

```
\addchap[Inhaltsverzeichniseintrag]{Überschrift}
\addchap*{Überschrift}
\addsec[Inhaltsverzeichniseintrag]{Überschrift}
\addsec*{Überschrift}
```

Für unnumerierte Abschnitte wie beispielsweise ein Vorwort oder Danksagungen stand bisher nur der `\chapter*`- bzw. `\section*`-Befehl zur Verfügung, die allerdings den Nachteil haben, daß sie zwar das bis auf die Numerierung komplette Überschrifts-Layout erzeugen, jedoch weder einen Eintrag ins Inhaltsverzeichnis noch einen Kolumnentitel generieren. Dies mußte, falls dies erwünscht war, nachträglich von Hand durchgeführt werden, was wohl nicht ganz der LaTeX-Philosophie entspricht.

Für die oben genannten Anwendungen wurden daher die neuen Befehle `\addchap` und `\addchap*` („additional chapter“, bei scrbook und scrreprt) sowie `\addsec` und `\addsec*` („additional section“, bei scrbook, scrreprt und scrartcl) implementiert, die bis auf die fehlende Numerierung genauso wie der `\chapter`- bzw. `\section`-Befehl (ohne Stern!) arbeiten, also alle Einträge an den entsprechenden Stellen erzeugen. Stern-Versionen `\addchap*` bzw. `\addsec*` sind den Stern-Versionen der üblichen Befehle sehr ähnlich, nur daß zusätzlich die Kolumnentitel gelöscht werden (geht z. B. der zu einem `\chapter*` gehörende Text über mehrere Seiten, so tauchen die – veralteten! – Kolumnentitel plötzlich wieder auf). Zusätzlich ist als Optionalargument wie bei den Originalen ein anderer Eintrag ins Inhaltsverzeichnis als die Abschnittsüberschrift angebbar.

3.6 Der „minisec“-Befehl

```
\minisec{Überschrift}
```

Manchmal ist eine Art Überschrift wünschenswert, die zwar etwas hervortritt (etwa durch die Wahl einer anderen Schrift), ansonsten aber eng mit dem nachfolgenden Text zusammenhängt, also nicht wie bei den üblichen Sektionierungsbefehlen relativ viel vertikalen Freiraum vor und nach der Überschrift läßt.

Wirkung des neuen „minisec“-Befehls:
Der Befehl `\minisec` („minisec“ steht für „mini-section“) bewirkt nun genau das oben genannte, indem er das Argument in `\sffamily\bfseries` mit einem vertikalen Freiraum von 1.5 ex vorher und keinem danach setzt. Die obige Zwischenüberschrift wurde als Beispiel mit `\minisec{Wirkung des ...}` gesetzt. Mini-Sections werden nicht mit ins Inhaltsverzeichnis aufgenommen. Da von vorneherein keine Numerierung vorgesehen ist, existiert auch keine Version mit Stern von diesem Befehl. Genauso wie `\marginline`,

\addchap, \addsec und die Titelseitenerweiterungen existiert \minisec nicht in der scrlettr-class.

3.7 Das „labeling“-Environment

```
\begin{labeling}[Trennzeichen]{längster Titel}
\item[Titel] ...
   ⋮
\end{labeling}
```

Wirkung: Wie in dieser tabellarischen Aufstellung, die auch gleichzeitig wieder ein Beispiel für die Wirkung des Environments ist, setzt `labeling` Text wie in einem normalen Absatz, nur daß zusätzlich ein Titel vorangestellt werden kann (in diesem Fall „Wirkung“).

Pro Titel sind, wie hier vorgeführt, mehrere Absätze möglich, um auch umfangreichere Texte übersichtlich gliedern zu können.

Syntax: `labeling` ist ein gewöhnliches Environment mit einer Zusatzangabe plus einem Optionalargument. Dabei ist {*längster Titel*} das längste \item-Argument inklusive allen eventuellen Schriftänderungsbefehlen. Wird kein Trennzeichen benötigt, so kann das optionale Argument [*Trennzeichen*] weggelassen werden.

Beispiel: Die Standardform eines mathematischen Beweises wird folgendermaßen erzeugt:

```
\begin{labeling}[:]{\sfb Auss.}
\item[\sfb Vor.] $x$ sei ...
\item[\sfb Auss.] Es gilt ...
\item[\sfb Bew.] Nach Satz ...
\end{labeling}
```

Dies bewirkt die Ausgabe:

Vor. : x sei ...

Auss.: Es gilt ...

Bew.: Nach Satz ...

4 Erweiterung der Internationalisierung

Werner Lemberg[1] bemängelte zurecht, daß die Standardnamen wie `\figurename` etc. nicht immer ausreichend sind, da manche Sprachen Zählwörter vor oder nach dem jeweiligen Zähler einfügen müssen (man denke da sogar im Deutschen an „Abbildung 1“ und „Erste Abbildung“). Desweiteren werden bei manchen Sprachen die Zwischenräume anders gesetzt bzw. sogar ganz unterdrückt.

Werner hat deshalb in seinem CJK-Paket[2] eine Unterteilung in drei Ebenen eingeführt. Um verschiedene Sprachen zu implementieren, kann jede dieser Ebenen getrennt verändert werden.

Ebene 1 besteht aus den hinlänglich bekannten Makros wie `\figurename` und alle weiteren Makros, die z. B. `german.sty` modifiziert. Desweiteren können die Makros `\sectfont` etc. etwaige zusätzliche Fonts aktivieren. Ebene 2 ist die zusätzliche Adaptierung von den Zählern selbst (`\thesection` könnte z. B. chinesische Schriftzeichen anstelle von arabischen Ziffern ansprechen). Die allgemeinste Form der dritten Ebene besteht aus den Makros `\chaptermarkformat`, `\sectionmarkformat`, `\subsectionmarkformat`, `\partformat`, `\chapterformat`, `\figureformat`, `\tableformat` und `\captionformat`, welche die Änderung der Reihenfolge von Zähler und Titel sowie die genaue Kontrolle der Zwischenräume etc. ermöglichen.

Für die gängigen Sprachen können alle Makros für die dritte Ebene identisch definiert werden. Bei Verwendung von „exotischen“ Sprachen, aber auch speziellen Darstellungen in den gängigen Sprachen können diese Makros umdefiniert werden.

Ab Version 2.2c (Versionsdatum [1995/05/25]) ist KOMA-Script explizit für die Verwendung von [CJK] vorbereitet[3]. Dies betrifft allerdings nicht `scrlettr.cls` wohl aber `scrpage.sty` (siehe Abschnitt 6).

[1] e-mail: `a7971428@unet.univie.ac.at`

[2] CJK steht für **C**hinesisch, **J**apanisch, **K**oreanisch.

[3] An dieser Stelle ein Dankeschön an Werner für die gute Zusammenarbeit.

5 Die scrlettr-class

5.1 Das Original

Immer mehr LaTeX-Anwender schreiben ihre Briefe ebenfalls mit TeX. Da jedoch die originale `letter`-class sehr auf das amerikanische Briefformat zugeschnitten ist, wird diese kaum verwendet. Damit nicht erst lange nach einer brauchbaren letter-class gesucht werden muß, ist eine solche ebenfalls im KOMA-Script-Paket enthalten.

Axel Kielhorn hat `scrlettr.cls` nach dem Vorbild von `script_l.sty` und den Gegebenheiten von LaTeX 2_ε erstellt. Ich selbst habe nur wenige, kleinere Änderungen vorgenommen, die mehr dazu dienten, die class meinen Wünschen anzupassen.

scrlettr ist nicht unbedingt die beste mögliche Wahl, genügt aber den meisten Ansprüchen und kann zumindest erheblich mehr als letter.

Eine typische Briefeingabedatei in der Standardklasse hat folgende Form:

```
\documentclass{letter}
\usepackage{german}
\name{Name des Absenders}
\address{Adresse des Absenders}
\signature{Unterschrift}
\begin{document}

\begin{letter}{Name und Adresse des Empf"angers
               des ersten Briefes}
\opening{Anrede (z.\,B. Sehr geehrte Damen und Herren,)}
Brieftext...Brieftext...Brieftext
\closing{Gru"sformel (z.\,B. Mit freundlichen Gr"u"sen)}
\ps{Postskriptum}
\cc{Verteiler (z.\,B. Direktion\\Personalb"uro)}
\encl{Anlagen (z.\,B. Lebenslauf\\Lichtbild)}
\end{letter}

\begin{letter}{Name und Adresse des Empf"angers
               des zweiten Briefes}
   ...
\end{letter}

   ...
\end{document}
```

Für scrlettr ist lediglich die Dokumentklasse `letter` eben durch `scrlettr` zu ersetzen.

Die Befehle `\signature`, `\closing`, `\ps`, `\cc` und `\encl` sind optional. Bei fehlender `\signature`-Angabe wird das `\name`-Argument als Unterschrift gesetzt. Ist eine gedruckte Unterschrift völlig unerwünscht, so kann diese durch `\signature{}` unterdrückt werden.

Zu beachten ist weiterhin, daß Adreßfeld, Datum usw. erst durch den `\opening`-Befehl erzeugt werden (wichtig beispielsweise bei Rechnungen ohne Anrede; dort muß dann trotzdem ein leerer `\opening{}`-Befehl stehen). Dies hat weiterhin zur Folge, daß beispielsweise bei einer zu langen `backaddress` (siehe unten) eine „Overfull hbox“ in der Zeile gemeldet wird, in der der `\opening`-Befehl steht, obwohl dieser gar nicht die eigentliche Ursache ist. Bei einer solchen Warnung sind also auch alle vorherigen Befehle zu prüfen!

Um sich bei jedem Brief die wiederholte Eingabe der eigenen Adresse und sonstiger Voreinstellungen zu ersparen, ist es empfehlenswert, diese in eine eigene Datei (beispielsweise `privat.sty`) zu schreiben, die dann einfach als package mit `\usepackage{privat}` angegeben wird. Ein Beispiel für eine solche Datei findet sich auf Seite 37.

5.2 Zusätzliche Eigenschaften der scrlettr-class

Die neue scrlettr-class ist gegenüber dem Original wesentlich erweitert und flexibler gestaltet worden. Sie zeichnet vor allem die Möglichkeit der Benutzung eigener Adreßdateien und nahezu uneingeschränkte Serienbrieffähigkeit aus.

5.2.1 Mehrsprachigkeit

Die scrlettr-class wurde von vorneherein mehrsprachig ausgelegt (Englisch, amerikanisches Englisch, Deutsch, Österreichisch, Französisch, Dank Simone Naldi neuerdings auch Italienisch und Dank Ralph J. Hangleiter neuerdings ebenfalls Spanisch; voreingestellt ist amerikanisches Englisch).

```
\captionsenglish
\captionsUSenglish
\captionsamerican
\captionsbritish
\captionsUKenglish
\captionsgerman
\captionsaustrian
\captionsfrench
\captionsitalian
\captionsspanish
```

Zur Einstellung der Sprache der automatisch gesetzten („Caption“-) Texte wie „Betreff“, „Seite“, „Anlagen“ usw. steht eine Reihe von Befehlen, zur Verfügung. Diese werden im allgemeinen von Sprachumschaltpaketen wie german oder Babel automatisch verwaltet. Notfalls können die Befehle aber auch direkt verwendet werden.

```
\dateenglish
\dateUSenglish
\dateamerican
\datebritish
\dateUKenglish
\dategerman
\dateaustrian
\datefrench
\dateitalian
\datespanish
```

Die Schreibweise des Datums wird durch die Befehle

`\dateenglish`	1/12/1993
`\dateUSenglish`	12/1/1993
`\dateamerican`	12/1/1993
`\datebritish`	1/12/1993
`\dateUKenglish`	1/12/1993
`\dategerman`	1.12.1993
`\dateaustrian`	1.12.1993
`\datefrench`	1.12.1993
`\dateitalian`	1.12.1993
`\datespanish`	1.12.1993

in der angegebenen Form festgelegt. Auch diese werden normalerweise direkt von Sprachumschaltpaketen verwaltet.

Es ist generell empfehlenswert (aber nicht zwingend notwendig), grundsätzlich german als zusätzliches package mit dem \usepackage-Befehl anzugeben, da dieses package den sehr komfortablen \selectlanguage-Befehl zur Verfügung stellt, der gleichzeitig Captiontexte, die Schreibweise des Datums, einige sprachspezifische Extras und Trennregeln der verwendeten Sprache anpaßt. Im Zusammenhang mit scrlettr ist jedoch zu beachten, daß \selectlanguage erst *nach* \begin{document} verwendet werden sollte.[1]

orgdate

Bei Verwendung des german oder Babel Pakets kann mit Hilfe der Klassenoption orgdate auch eine Datumsdefinition innerhalb der scrlettr-Klasse verhindert werden. Damit wird dann das Datum in der Form verwendet, in der es in den genannten Paketen definiert wird.

Beispiel 1: Für eine Datei mit rein englischsprachigen Briefen (amerikanisches Englisch) genügt:

```
\documentclass{scrlettr}
\begin{document}
\begin{letter}{...}
...
```

Beispiel 2: Für eine Datei mit rein deutschsprachigen Briefen genügt:

```
\documentclass{scrlettr}
\usepackage{german}
\begin{document}
\begin{letter}{...}
...
```

Beispiel 3: Bei gemischtsprachigen Briefen empfiehlt sich:

```
\documentclass{scrlettr}
\usepackage{german}
\begin{document}

\selectlanguage{USenglish}
\begin{letter}{...}
```

[1] Die briefspezifischen Caption-Texte und Datumsformate werden nach einem Vorschlag von Sven Gohlke erst durch den \begin{document}-Befehl definiert. Daher werden diese erst nach diesem Befehl aktiv. Durch eine spezielle Anpassung an german.sty Version 2.5b ist die Sprachauswahl nicht mehr von festen Sprachnummern abhängig. Dafür wird jedoch gefordert, daß die Spachnummern zu german.sty kompatibel definiert sind.

```
...
\end{letter}

\selectlanguage{german}
\begin{letter}{...}
...
\end{letter}

...
```

5.2.2 Zweiseitiger Druck

`twoside`

Zweiseitig gedruckte Briefe werden durch Angabe der Option `twoside` unterstützt; jedoch ändert sich hier im Gegensatz zu den übrigen KOMA-Script-classes der Satzspiegel nicht, sondern es wird lediglich sichergestellt, daß ein neuer Brief immer auf einer ungeraden *Druck*seite beginnt. Die *interne* TEX-Seitenzählung startet bei jedem `\begin{letter}`-Befehl wieder bei 1.

Darüber hinaus wird eine Warnung ausgegeben, um darauf hinzuweisen, daß es sich nicht wirklich um ein zweiseitiges Layout handelt.

5.2.3 Layout

Die `scrletter`-class verwendet einen festen Satzspiegel, an dem nichts mehr verändert werden sollte (Ausnahmen siehe unten). Er ist recht anspruchsvoll, was Maßhaltigkeit und bedruckbaren Bereich des verwendeten Druckers angeht, da Falzmarken so weit links wie möglich und die Adresse wirklich im Adreßfenster eines Fensterbriefumschlages stehen sollen. Aus diesem Grund sollte die Einstellung des Druckertreibers mit der LATEX 2_ε-Testseite überprüft werden.

Sollten die Falzmarken auf dem Drucker nicht erscheinen, so ist zunächst zu überprüfen, ob sie überhaupt eingeschaltet sind. Ist dies der Fall, so hat der Drucker vermutlich einen linken Rand von mehr als 5 mm. Anderenfalls müßte zumindest ein Teil der Falzmarken zu sehen sein. In diesem Fall bleibt nur die dokumentierte Länge `\foldhskip` zu erhöhen. Man kann dies beispielsweise in der Präambel oder im Rahmen einer Datei tun, in der man seinen eigenen Briefkopf definiert (siehe Abschnitt 5.2.8).

Die Höhe des Kopfbereichs der Seite ist in der Version 2.0 von 20 mm auf 22 mm vergrößert worden[2]; damit gibt es hoffentlich keine Probleme mehr bei Verwendung des

[2]Dadurch ist der Abstand des oberen Papierrandes vom Briefkopf von 10 mm auf 9 mm geschrumpft; da einige Drucker damit bereits Probleme hatten, habe ich den oberen Rand von 15,5 mm auf 19 mm vergrößert. Dafür wurde wiederum der Abstand zwischen Kopfzeile und Text von 9 mm auf 8 mm

voreingestellten Briefkopf-Formates[3]. Der Fußbereich der Seite ist 25 mm hoch, linker und rechter Rand sind beide 25,4 mm breit.

Absätze werden ohne Einzug und mit einem vertikalen Abstand von einem halben `\baselineskip` gesetzt. Ein unterer Randausgleich wird auch im zweiseitigen Druck nicht durchgeführt (`\raggedbottom`).

Gegenüber der Version 1.0 von `script_l.sty` wurde die Brieftext-Zentrierung leicht verbessert, was die Probleme bei umfangreicheren Briefen mit sehr langen Absätzen hoffentlich beseitigt hat.

Veränderung des Satzspiegels in Ausnahmefällen

Sollte sich *nach* Überprüfung und Korrektur der Drucker- bzw. Treibereinstellung ergeben, daß die komplette Briefseite doch nicht gedruckt werden kann, so können die Einstellungen von `\textheight` und `\footskip` geändert werden. Durch eine Verkleinerung von `\textheight` wird die Höhe des Textbereichs kleiner, dadurch wird der untere Rand (unter dem Fuß) größer. Durch eine Verkleinerung von `\footskip` rückt der Fuß näher an den Textbereich, wodurch der untere Rand (unter dem Fuß) ebenfalls gröser wird. In seltenen Fällen muß außerdem `\topmargin` vergrösert werden. In diesem Fall ist unbedingt `\textheight` um denselben Wert zu verkleinern, weil sonst der Seiteninhalt bzw. der Fuß unten aus dem Papier rutscht.

Die Änderung kann entweder in der Dokumentpräambel, in der Datei mit dem persönlichen Briefkopf oder in `scrlettr.cls` selbst erfolgen. Bitte beachten Sie jedoch, daß die veränderte Datei nicht weitergegeben werden darf und Briefe dann bei Bearbeitung mit Original scrlettr anders umbrochen werden!

5.2.4 Adreßfeld und Zusatz-„location"-Feld

Zur Beschriftung des Briefumschlages gab es im Original den `\makelabels`-Befehl, der dazu gedacht war, Adreßetiketten zu bedrucken. Da mir keine Konfiguration bekannt ist, in der dieser Befehl mit all seinen Anforderungen bezüglich Papierwechsel etc. wirklich Sinn macht, habe ich ihn erst gar nicht in die neue Brief-class miteinbezogen (mal ehrlich: Haben Sie `\makelabels` benutzt?) Hier sehe ich auch die — hoffentlich — einzige Möglichkeit, wo eine Inkompatibilität zu den Original-classes auftreten kann.

Als Entschädigung ermöglicht scrlettr die Verwendung von Fensterbriefumschlägen, die Adreßetiketten überflüssig machen. In einem Feld der Höhe `\addrheight` und Breite `\addrwidth`, das `\addrvskip` von der Textbereichsoberkante und (`\addrindent` plus 25,4 mm) vom linken Rand des Papiers entfernt ist, wird dazu die bei `\begin{letter}` angegebene Adresse vertikal zentriert.

verringert. Die entsprechende Zeile ist dokumentiert.

[3] Diese Probleme traten auf bei Verwendung der 12pt-Schrift und wenn beide Zeilen der Adresse Unterlängen besaßen. Dreizeilige Adressangaben sind jedoch nach wie vor mangels Platz problematisch aber möglich. Falls mehr Information im Briefkopf untergebracht werden soll, so muß ein eigener entwickelt werden.

`\backaddress{`*Rücksende-Adresse*`}`
`\specialmail{`*Versandart*`}`

Optional kann noch mit `\backaddress` eine kleingeschriebene Rücksende-Adresse und mit `\specialmail` eine spezielle Versandart (beispielsweise „Einschreiben“) in dieses Feld eingeblendet werden.

`\location{`*Zusatz-Information*`}`

Im freien Raum rechts neben dem Adreßfeld läßt sich noch zusätzlicher Text (beispielsweise die Zweigstellen einer Firma) unterbringen. Dieser wird durch `\location` in ein `\locwidth` breites Feld geschrieben, dessen rechter Rand mit der rechten Begrenzung des Textbereiches übereinstimmt.

`\addrfieldoff`
`\addrfieldon`

Beide Felder zusammen sind durch `\addrfieldoff` ab- bzw. mit `\addrfieldon` wieder anschaltbar (Default: `\addrfieldon`). Dies kann z.B. für persönliche Schreiben ohne Geschäftsbriefcharakter von Nutzen sein (wenngleich ich in diesem Falle eher zum Füller greifen würde). Hier wurden Befehle an Stelle von Optionen verwendet, da es somit möglich ist, in einem Dokument mehrere Briefe mit unterschiedlichen Einstellungen unterzubringen.

Noch eine kleine Anmerkung zur Schreibweise des Wortes „Adresse“: Im Deutschen schreibt es sich mit *einem* „d“, während es im Englischen „a*dd*ress“ heißt. Daher ist bei allen Befehlen, die „address“ enthalten, auf das doppelte „d“ zu achten. Anweisungen, die etwas mit dem unten beschriebenen `.adr`-File zu tun haben, werden allerdings mit *einem* „d“ geschrieben (`\adrentry` und `\adrchar`).

`wlocfield`
`slocfield`

Die Breite des Adreß-Feldes ist normalerweise auf 70 mm beschränkt. Für das Zusatzfeld daneben (`\location`) steht in der Grundeinstellung die Hälfte der restlichen Textbreite zur Verfügung. Da mir selbst dies etwas zu wenig war, existiert eine Option `wlocfield` („wide location field“), mit der 2/3 der restlichen Textbreite verwendet werden. Dies sollte bei der Mehrzahl der Fensterumschläge problemlos sein. Natürlich existiert auch eine gegenteilige Option `slocfield` („small location field“), mit der die Standardeinstellung 1/2 verlangt wird.

5.2.5 Referenzfeld

```
\yourref{Ihr Zeichen}
\yourmail{Ihr Schreiben vom}
\myref{Unser Zeichen}
\customer{Kundennummer}
\invoice{Rechnungsnummer}
\refitemi{Eigenes Feld 1}
\refitemii{Eigenes Feld 2}
\refitemiii{Eigenes Feld 3}
```

Unterhalb des Adreßfeldes steht normalerweise das Datum bzw. bei Geschäftsbriefen ein Referenzfeld mit vorgedruckten Unterfeldern für Kundennummer, Durchwahl, Sachbearbeiter und ähnliches. Dies ist jetzt auch möglich: Ohne weitere Angabe erscheint rechtsbündig Ort (definiert durch `\place{`*Ort*`}`) und Datum, während bei Benutzung mindestens eines der obenstehenden Befehle das Datum in Vordruckform und die gewünschten Unterfelder horizontal gegeneinander zentriert gesetzt werden.

```
\refitemnamei{Vordruck}
\refitemnameii{Vordruck}
\refitemnameiii{Vordruck}
```

Die „Vordrucke“ der selbstdefinierten Unterfelder lassen sich mit diesen Befehlen bestimmen.

5.2.6 Titel und Betreff

```
\title{Überschrift}
```

Keine eigene Titelseite, sondern eine fettgedruckte Überschrift (beispielsweise „Rechnung“, „1. Mahnung“) wird durch den Befehl `\title` unterhalb des Referenzfeldes erzeugt.

```
\subject{Kurzinhalt des Briefes}
\subjecton
\subjectoff
```

Ein „Betreff“ läßt sich mit `\subject` angeben, wobei mit `\subjecton` bzw. `\subjectoff` das Wort „Betr.“ selbst an- und ausgeschaltet werden kann (voreingestellt ist aus, also `\subjectoff`).

5.2.7 Seitenstile

```
\firsthead{Kopf}
\firstfoot{Fuß}
\nexthead{Kopf}
\nextfoot{Fuß}
```

In der scrlettr-class stehen nach wie vor die Seitenstile `empty`, `plain` und `headings` zur Verfügung (Default ist `plain`). Zusätzlich gibt es noch speziell für die erste Seite eines Briefes den Pagestyle `firstpage`. Der eigentliche Clou an der Sache ist nun, daß die Einträge in den Kopf- und Fußbereich der Seiten bei `firstpage` und `headings` frei gestaltbar sind. Dies geschieht durch die Befehle `\firsthead`, `\firstfoot`, `\nexthead` und `\nextfoot`, die bei einer Änderung auf jeden Fall *vor* dem Aufruf des zugehörigen `\pagestyle{...}`- bzw. `\begin{letter}`-Befehls ausgeführt werden müssen. Als Beispiel sollen die Vorbelegungen der class-Datei dienen[4]:

```
\firsthead{\parbox[b]{\textwidth}{\begin{center}
           {\sc \ignorespaces \fromname}\\[-8pt]
           \rule{\textwidth}{.4pt}\\
           \ignorespaces \fromaddress \end{center}}}
\firstfoot{}
\nexthead{\parbox[b]{\textwidth}{{%
          \sc \ignorespaces \fromname}\\[8pt] \headtoname\
          \ignorespaces \toname \hfill \datename:\ \today
          \hfill \pagename\ \thepage\\[-8pt]
          \rule{\textwidth}{.4pt}}}
\nextfoot{}
```

Dadurch ist es nun möglich, eigene Briefköpfe und Fußleisten (bespielsweise für Bankverbindungen, Gerichtsstand etc.) zu entwerfen. Der Phantasie sind eigentlich nur durch die zur Verfügung stehenden Zeichensätze Grenzen gesetzt; Ausgefallenes läßt sich durch das `picture`-Environment oder „Metafont“[5] realisieren.

5.2.8 Adreßdateien

```
\adrentry{Name}{Vorname}{Adresse}{Telefonnummer}{}{}{Kommentar}{Kürzel}
```

Um nicht in jedem Brief die komplette Anschrift des Adressaten nachschlagen und vor allem eingeben zu müssen, wurden Adreßdateien (`.adr`-Files) eingeführt. Sie enthalten

[4] Das Verständnis der Definitionen ist zugegebenermaßen nicht ganz einfach. Die Bedeutung der verwendeten Variablen schlagen Sie bitte in der Variablenübersicht in Abschnitt 5.3 nach.

[5] Das ist keineswegs so schwer, wie es zunächst erscheint, da es bereits Vektor-Graphikprogramme gibt, die auch Metafont-Ausgaben beherrschen. Aber Vorsicht! Metafont macht süchtig!

für jeden Adressaten einen Eintrag in Form eines Befehls `\adrentry`, dessen fünftes und sechstes Argument dem Anwender zur freien Verfügung steht.

Eine solche Adreßdatei bindet man mit `\input` entweder direkt in den Quelltext des Briefes (am besten noch in der Präambel) oder in das oben vorgeschlagene Optionsfile, das die Voreinstellungen enthält, ein. Im letzteren Fall kann man statt `\input` natürlich auch den Befehl `\InputIfFileExists` verwenden, der eine Fehlerbehandlung für den Fall erlaubt, daß die Datei nicht existiert.

Beispiel 1: Frank Neukams persönliches Optionsfile sieht folgendermaßen aus:

```
% privat.sty
\name{Frank Neukam}
\address{Franz-Stadelmayer-Str. 30\\97074 W"urzburg}
\backaddress{F. Neukam, F.-Stadelmayer-Str. 30, 97074 W"urzburg}
\place{W"urzburg, }
\input{privat.adr}
```

Beispiel 2: Markus Kohms persönliches Optionsfile ist „etwas“ aufwendiger, demonstriert aber gut die tatsächlichen Möglichkeiten:

```
% mjk.sty, private Einstellungen f"ur scrlettr
% (in Verbindung mit der Option 'wlocfield' verwenden)
% Markus Kohm, irgendwann 1994 und in Wirklichkeit l"angst "uberholt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\ProvidesFile{mjk}
\typeout{Absender ist: Markus Kohm}
\font\mklogo  =mklogo scaled \magstep0 % beinhaltet mein Logo

\newcommand\mk{\mklogo\symbol{0}}
\name{Markus Kohm}
\signature{\qquad Markus Kohm}
\address{Fichtenstra"se 63\\68535 Edingen-Neckarhausen}
\backaddress{Markus Kohm $\cdot$\
             Fichtenstra"se 63 $\cdot$\
             68535 Edingen-Neckarhausen}
\place{Karlsruhe, }
\telephone{(07\,21)~} % Geheimnummer entfernt
\location{\begin{flushleft}\small\sf%
          {\sfb email:}\\Mausnet:\\\hspace{1em}Markus Kohm @ HD\\[.8ex]
          Internet:\\\hspace{1em} Markus\_Kohm@hd.maus.de\\[1.5ex]
          {\sfb Telefon:}~~\telephonenum\\
          \end{flushleft}}
\firsthead{\parbox{\textwidth}{%\vspace*{1ex}
           \parbox{2cm}{\mk}\hfill\parbox{\locwidth}{\begin{flushleft}
           \fontsize{12}{14.5}\selectfont
```

```
                {\sfb\ignorespaces\fromname}\\
                {\sf\ignorespaces\fromaddress}\\
                \end{flushleft}}\par\vspace{1ex}\rule{\textwidth}{0.8pt}}}
        \firstfoot{\parbox[t]{\textwidth}{\begin{center}
                \sffamily\slshape\fontsize{8}{9}\selectfont
                \rule{\textwidth}{.4pt}\\
                Bankverbindung: Sparda--Bank Karlsruhe (BLZ~660\,905\,00)
                Konto--Nr.~995\,894\end{center}}}
        \nexthead{\parbox[b]{\textwidth}{{%
                \sfb\ignorespaces\fromname}
                \hfill\sf\headtoname\ \ignorespaces\toname
                \hfill\datename:\ \@date\hfill\pagename\
                \thepage\\[-8pt]\rule{\textwidth}{.4pt}}}
        \nextfoot{\parbox[t]{\textwidth}{\begin{center}
                \sffamily\slshape\fontsize{8}{9}\selectfont
                \rule{\textwidth}{.4pt}\\
                Bankverbindung: Sparda--Bank Karlsruhe (BLZ~660\,905\,00)
                Konto--Nr.~995\,894\end{center}}}
        \pagestyle{headings}
        \InputIfFileExists{privat.adr}{\typeout{Private Adressen geladen}}
                                      {\typeout{Keine privaten Adressen!}}
        \endinput
```

Für jeden `.adr`-Eintrag wird dann die komplette Anschrift unter dem jeweiligen Kürzel als TEX-Makro abgelegt. Um das Überschreiben von TEX- oder LATEX-Befehlen zu vermeiden, ist es empfehlenswert, für Kürzel nur Großbuchstaben zu verwenden. Lautet also beispielsweise ein `.adr`-Eintrag

```
\adrentry{abc-Lernsysteme}
        {}
        {z.Hd. Frau Streb\\Schulstr.1\\2345 Paukstadt}
        {0\,678 / 90\,12\,34}
        {}
        {}
        {}
        {ABC}
```

so schreibt man anstelle der kompletten Adresse nur noch `\begin{letter}{\ABC}`. Mit Hilfe der im SCRIPT-Paket enthaltenen `dir.tex`-Datei kann aus einer `.adr`-Datei ein Adreßheft erzeugt werden, das nach Zurechtschneidung in die Standard-Taschenkalenderhüllen paßt. Dazu sollten die Einträge alphabetisch sortiert sein und vor jedem neu begonnenen Anfangsbuchstaben des Nachnamens ein `\adrchar{`*Anfangsbuchstabe*`}` stehen. Von der scrlettr-class wird dieser Befehl ignoriert; `dir.tex` jedoch beginnt bei einer solchen Anweisung eine neue Seite und schreibt diesen Buchstaben in die Kopfzeile. Am besten probieren Sie dieses Programm einfach einmal anhand einer kleinen Adreßdatei aus, alle benötigten Informationen werden während des TEX-Kompilerlaufes erfragt.

Ähnlich wie `dir.tex` arbeitet das Programm `phone.tex`, nur daß dieses anstelle eines Adreßheftes ein Telefonnummern-Verzeichnis anlegt. `\adrchar` erzeugt hier lediglich einen kleinen vertikalen Zwischenraum.

5.2.9 Serienbriefe

Eine weitere Anwendung der `.adr`-Dateien sind Serienbriefe.

Beispiel: Mit der folgenden kurzen Datei kann eine Einladung an sämtliche Kollegen eines Betriebes erzeugt werden:

```
\documentclass{scrlettr}
\usepackage{privat,german}
\begin{document}
\def\adrentry#1#2#3#4#5#6#7#8{
 \begin{letter}{#2 #1\\#3}
 \opening{Liebe Kolleginnen und Kollegen,}
 anl"a"slich meines ...
 \closing{Mit freundlichen Gr"u"sen}
 \end{letter}
}
\input{kollegen.adr}
\end{document}
```

Soll der Brieftext abhängig von Adressatenmerkmalen, die in den beiden noch freien Argumenten des `\adrentry`-Befehls stehen, veränderlich sein, können Konstrukte der folgenden Form verwendet werden[6]:

```
\def\adrentry#1#2#3#4#5#6#7#8{
 \ifcase #6
  % Dieser Zweig wird bei #6 = 0 abgearbeitet.
  % In diesem Beispiel sind Empf"anger ohne Kinder uninteressant,
  % weshalb im \else-Zweig nur Briefe an Kunden mit mindestens
  % einem Kind erzeugt werden.
 \else
  % #6 > 0
  \begin{letter}{#2 #1\\#3}
   \if #5m \opening{Sehr geehrter Herr #1,} \fi
   \if #5w \opening{Sehr geehrte Frau #1,} \fi

   Sie als% Direkt nach 'als' mu"s ein % stehen, da der \if-Befehl
          % vor 'gl"ucklich..' einen zus"atzlichen Leerraum
          % generiert (nach \fi jedoch nicht}.
```

[6] Als Beispiel ist das Geschlecht des Empfängers im Argument #5 kodiert: m für männlich, w für weiblich. In Argument #6 steht die Anzahl der Kinder.

```
    \if #5m gl"ucklicher Vater \fi
    \if #5w gl"uckliche Mutter \fi
    wissen sicher am besten, was gut f"ur
    \ifnum #6 < 2
     Ihr Kind
    \else
     Ihre Kinder
    \fi
    ist. Um Sie...

    \closing{Mit freundlichen Gr"u"sen}
   \end{letter}
  \fi
}
```

Es ist also auch möglich, den Brieftext auf bestimmte Empfängermerkmale gezielt abzustimmen und so den Eindruck eines persönlichen Schreibens zu erwecken. Die Anwendungsbreite ist lediglich durch die maximale Anzahl von zwei freien `\adrentry`-Argumenten begrenzt.

5.2.10 Das scraddr-package zur Auswertung von Adreßdateien

Dieses Paket liest beliebige - auch mehrere - Adreßdateien im oben beschriebenen Format ein und legt für jeden Eintrag eine Reihe von Makros an. Bedingung dafür ist, daß das achte Element jedes `\adrentry`-Eintrags - der Kürzeleintrag - nicht leer ist. Im Falle von identischen Kürzeleinträgen überschreiben spätere Einträge frühere. Die Inhalte können dann über spezielle Befehle ermittelt werden.

Das Paket besitzt keine Optionen.

`\InputAddressFile{`*Dateiname*`}`

Mit Hilfe des Befehls `\InputAddressFile` wird die angegebene Adreßdatei eingelesen und in Adreßmakros gewandelt, die dann beliebig verwendet werden können. Die Adreßmakros werden global generiert. Andere Makros insbesondere die Definition von `\adrentry` bleiben davon unberührt.

Existiert die Datei *Dateiname* nicht, so wird ein Fehler ausgegeben, ansonsten wird die Ladeaktion angezeigt.

```
\Name{Kürzel}
\FirstName{Kürzel}
\LastName{Kürzel}
\Address{Kürzel}
\Telephone{Kürzel}
\FreeI{Kürzel}
\FreeII{Kürzel}
\Comment{Kürzel}
```

Da keine Vorschriften existieren, wie genau das Kürzel aufgebaut sein muß, können innerhalb des Kürzels theoretisch auch solche Zeichen enthalten sein, die für Makronamen normalerweise ungeeignet sind (beispielsweise Punkte oder Ziffern). Deshalb existiert ein Satz von Befehlen, über den auf die zu einem Kürzel gehörenden Elemente zugegriffen werden kann. Dabei wird davon ausgegangen, daß die Adreßdatei dem zuvor vorgeschlagenen Aufbau entspricht.

Bedingung für die korrekte Funktion ist, daß die `\adrentry`-Einträge in der Adreßdatei dem zuvor genannten Aufbau entsprechen! Außerdem darf in den Kürzeln kein Zeichen enthalten sein, das u.U. von einer Makroexpansion betroffen wäre. Es empfiehlt sich daher, sich auf Buchstaben (ohne Umlaute!) und Ziffern zu beschränken.

5.2.11 Adreßdatenbankkonverter

Gerd Neugebauer hat für BibTeX eine Adreßdatenbank definierter und dazu passend den Bibliographie-Stil `address.bst` erstellt. Diese Datenbank enthält wesentlich mehr Einträge als die Adreßdateien, die Script verwendet. Darüber hinaus ist sie beliebig erweiterbar. Es drängt sich natürlich die Frage auf, ob man nicht irgendwie die BibTeX-Datenbank bzw. die relevanten Teile daraus für Script aufbereiten und verwenden kann.

Axel Kielhorn ist dieses Problem angegangen und hat zwei Bibliographie-Stile geschrieben, die als Beispiele dienen können. Diese Stile stellen keine ultimative Lösung dar, sondern sie sollen Hinweise geben, was man selbst alles machen kann. Durch Abänderung der Bespiele kann man leicht seinen eigenen Konverter erzeugen. Dadurch kann man z. B. beliebige Felder der BibTeX-Datenbank in die freien Felder der Adreßdatei übernehmen.

Zur Funktion von BibTeX lesen Sie bitte in der Anleitung zu BibTeX (`btxdoc` und `btxhak`) nach.

Die Bibliographie-Stile zur Konvertierung

Inzwischen liegt dem Paket auch die Originalanleitung von Axel Kielhorn, `adrguide.tex`, bei, deren Studium ich Ihnen ebenfalls nahe lege.

Der erste Bibliographie-Stil, `addrconv.bst`, erzeugt einfach eine Adreßdatei mit den notwendigen Einträgen *Name*, *Vorname*, *Adresse* und *Telefonnummer*. Dabei werden die

Einträge der Adreßdatei alphabethisch sortiert und auch der `\adrchar` wird automatisch gesetzt.

Den Inhalt der Adreßdatei kann man anschließend sehr schön mit den beiden Beispieldateien `dir.tex` und `phone.tex` ausgeben lassen.

Der zweite Bibliographie-Stil, `birthday.bst`, erzeugt stattdessen eine Geburtstagsliste. Dabei werden die Einträge nicht alphabetisch sondern nach den Geburtstagen (also Monat und Jahr) sortiert.

Eine sinnvolle Ausgabe ist hier ebenfalls mit `dir.tex` möglich.

Eine Liste der E-Mail-Adressen kann ebenfalls erstellt werden.

Die Anwendung der Bibliographie-Stile zur Konvertierung

Normalerweise schreibt man die Information, welcher Bibliographie-Stil und welche -Datenbank verwendet werden soll in die LaTeX-Datei, in die die Datenbank eingebunden werden soll. Beim ersten TeX-Lauf wird die entsprechende Information dann in der zugehörigen `aux`-Datei abgelegt. Von dort liest BibTeX die Informationen und erzeugt dann mit Hilfe des Bibliographie-Stils in einer `bst`-Datei aus einer Quell-`bib`-Datei die Zieldaten in der `bbl`-Datei. Diese werden dann beim nächsten TeX-Lauf mit eingelesen und verarbeitet.

In unserem Fall der Konvertierung ist das nur bedingt möglich. So existiert zwar eine `bib`-Datei und auch die beiden Bibliographie-Stile in den `bst`-Dateien, aber es fehlt die `tex`-Datei, aus der LaTeX die Information in eine `aux`-Datei schreiben könnte. Genau dafür dienen die beiden Dateien `addrconv.tex` und `birthday.tex`. Es handelt sich dabei um interaktive TeX-Programme, die sowohl unter TeX(PlainTeX) als auch unter LaTeX funktionieren.

Nach dem Start wird man nach dem Namen der BibTeX-Datenbank gefragt, die konvertiert werden soll. Der Name muß ohne die Endung „`.bib`“ eingegeben werden! Daraufhin wird eine Datei gleichen Namens aber mit der Endung „`.aux`“ erzeugt. Starten Sie nun BibTeX, wobei Sie als Parameter ebenfalls wieder den Dateinamen ohne Endung angeben. BibTeX erzeugt daraufhin (vorausgesetzt die `bst`-Datei wird gefunden) eine Datei gleichen Namens jedoch mit der Endung „`.bbl`“. Jetzt müssen Sie nur noch die Endung in „`.adr`“ ändern und können die Datei z.B. mit `dir.tex` oder `phone.tex` weiterbearbeiten.

`addrconv.tex` erzeugt eine `aux`-Datei für die Verwendung von `addrconv.bst`, während `birthday.tex` eine `aux`-Datei für die Verwendung von `birthday.bst` und `email.tex` eine `aux`-Datei für die Verwendung von `email.bst` erzeugt.

Hier das Protokoll einer Beispielbearbeitung für die Beispieldatenbank `example.bib`:

```
sh>tex addrconv.tex
This is TeX, Version 3.1415926536
(addrconv.tex
```

```
Now you have to typein the name of the BibTeX addressfile, you want to
convert to script-address-file-format (without extension '.bib'):
Geben Sie nun den Namen der BibTeX-Adressdatei ein, die sie in das
Script-Adressdateiformat konvertieren wollen (ohne '.bib'):

addressfile=example

After running BibTeX rename file 'example.bbl' to 'example.adr'!
Nach dem BibTeX-Lauf benennen Sie bitte die Datei 'example.bbl' in
'example.adr' um!
 )
No pages of output.
Transcript written on addrconv.log.
sh>bibtex example
This is BibTeX, 8-bit Version 0.99c // 0.6cs
The top-level auxiliary file: example.aux
The style file: addrconv.bst
Database file #1: example.bib
sh>mv example.bbl example.adr
```

Die erzeugte Adreßdatei, `example.adr`, sieht dann so aus:

```
\adrchar{K}
\adrentry{Kielhorn}{Axel}
{Stra{\ss}e des 16.~Mai 17 \\ 38118 Braunschweig}{0531 / 89 34 39}{}{}{}{}
\adrentry{Kielhorn}{Ralf}
{Stra{\ss}e des 17.~Juni 17 \\ 38118 Braunschweig}{0531 / 89 34 39}{}{}{}{}
\adrentry{Kohm}{Markus}
{Augartenstra{\ss}e 27 \\ 76137 Karlsruhe}{}{}{}{}{}

\adrchar{M}
\adrentry{Mustermann}{Hans}
{Einbahnstra{\ss}e 1 \\ 01234 Heimstatt}{01234 / 5 67 89}{}{}{}{}
```

Sollten Sie Fragen zu den Bibliographie-Stilen haben, so wenden Sie sich bitte direkt an Axel Kielhorn, seine eMail-Adresse finden Sie in `example.bib`.

5.3 Befehls- und Variablenübersicht

Briefspezifische Befehle, die strukturbeschreibend sind oder eine Ausgabe erzeugen:

`\begin{letter}{`*Adressat*`}`	Markiert den Beginn eines Briefes an *Adressat* und beginnt eine neue Seite
`\end{letter}`	Markiert das Ende eines Briefes
`\opening{`*Anrede*`}`	Setzt alle Teile eines Briefes oberhalb und einschließlich der *Anrede*
`\closing{`*Grußformel*`}`	Setzt *Grußformel* und Unterschrift
`\ps{`*Postskriptum*`}`	Setzt ein *Postskriptum*
`\cc{`*Verteiler*`}`	Setzt eine Verteilerliste (Einträge durch `\\` abgetrennt)
`\encl{`*Anlagen*`}`	Setzt eine Anlagenliste (Einträge durch `\\` abgetrennt)

Befehle der Adreßdateien:

`\adrchar` und `\adrentry` (s.o.)

Befehle zur Sprachumschaltung:

`\captionsenglish`	Selektiert englische Caption-Texte
`\captionsUSenglish`	Selektiert amerikanische Caption-Texte
`\captionsgerman`	Selektiert deutsche Caption-Texte
`\captionsaustrian`	Selektiert österreichische Caption-Texte
`\captionsfrench`	Selektiert französische Caption-Texte
`\captionsitalian`	Selektiert italienische Caption-Texte
`\dateenglish`	Englisches Datum (1/12/1993)
`\dateUSenglish`	Amerikanisches Datum (12/1/1993)
`\dategerman`	Deutsches Datum (1. 12. 1993)
`\dateaustrian`	Österreichisches Datum (1. 12. 1993)

`\datefrench`	Französisches Datum (1. 12. 1993)
`\dateitalian`	Italienisches Datum (1. 12. 1993)

Sprachabhängige Variablen

(dürfen an beliebiger Stelle nach dem `\begin{document}`-Befehl verwendet werden, können aber nur mit Hilfe von `\renewcommand` geändert werden; die Liste enthält die Variableninhalte nur in Englisch, Deutsch und Französisch, da die amerikanischen Namen den englischen und die österreichischen den deutschen entsprechen; die Inhalte für weitere Sprachen – z. B. das ebenfalls implementierte Italienisch – entnehmen Sie bitte der Implementierungsdokumentation bzw. der Klasse):

`\yourrefname`	Your ref. / Ihr Zeichen / Vos références
`\yourmailname`	Your letter of / Ihr Schreiben vom / Votre lettre du
`\myrefname`	Our ref. / Unser Zeichen / Nos références
`\customername`	Customer no. / Kundennummer / Numéro de client
`\invoicename`	Invoice no./ Rechnungsnummer / Numéro de facture
`\subjectname`	Subject / Betr. / Concernant
`\ccname`	cc / Kopien an / Copie à
`\enclname`	encl / Anlagen / Annexes
`\headtoname`	To / An / A
`\datename`	Date / Datum / Date
`\pagename`	Page / Seite / Page

Briefspezifische Variablen und deren Befehle zur Neu- oder Umdefinierung

(Variablen mit `@` sind geschützt; alle anderen können an beliebiger Stelle verwendet werden):

`\name`	ändert `\fromname`
`\signature`	ändert `\fromsig`
`\address`	ändert `\fromaddress`

`\place`	ändert	`\fromplace`
`\location`	ändert	`\fromlocation`
`\backaddress`	ändert	`\frombackaddress`
`\telephone`	ändert	`\telephonenum`
`\yourref`	ändert	`\varyourref`
`\yourmail`	ändert	`\varyourmail`
`\myref`	ändert	`\varmyref`
`\customer`	ändert	`\varcustomer`
`\invoice`	ändert	`\varinvoice`
`\refitemi`	ändert	`\varrefitemi`
`\refitemii`	ändert	`\varrefitemii`
`\refitemiii`	ändert	`\varrefitemiii`
`\refitemnamei`	ändert	`\varrefitemnamei`
`\refitemnameii`	ändert	`\varrefitemnameii`
`\refitemnameiii`	ändert	`\varrefitemnameiii`
`\specialmail`	ändert	`\@specialmail`
`\title`	ändert	`\@title`
`\subject`	ändert	`\@subject`
`\firsthead`	ändert	`\@firsthead`
`\firstfoot`	ändert	`\@firstfoot`
`\nexthead`	ändert	`\@nexthead`
`\nextfoot`	ändert	`\@nextfoot`

Briefspezifische Längenangaben

(voreingestellte Länge in eckigen Klammern):

`\foldhskip`	Abstand der Falzmarken vom linken Papierrand [3.5 mm]
`\foldvskipi`	Abstand der ersten Falzmarke vom oberen Papierrand [62 mm]
`\foldvskipii`	Abstand der zweiten Falzmarke von der ersten Falzmarke [40.5 mm]
`\foldvskipiii`	Abstand der dritten Falzmarke von der zweiten Falzmarke [58.5 mm]
`\addrvskip`	Abstand des Adreßfensters von der Textbereichs-Oberkante [7.5 mm]
`\addrindent`	Abstand des Adreßfensters vom linken Rand des Textbereiches [0 mm]
`\addrwidth`	Breite des Adreßfensters [70 mm]
`\addrheight`	Höhe des Adreßfensters [35 mm]
`\locwidth`	Breite des „Location“-Feldes [(`\textwidth−\addrwidth`)/2 bei Option `slocfield` oder (`\textwidth−\addrwidth`)∗2/3 bei Verwendung der Option `wlocfield`]
`\refvskip`	Abstand des Referenzfeldes von der Adreßfeld-Unterkante [5.5 mm]
`\sigindent`	Abstand der Grußformel und der Unterschrift vom linken Rand des Textbereiches [0 mm]

Schalter

(an jeden Schalternamen ist noch `on` bzw. `off` anzuhängen):

`\foldmarks` schaltet Falzmarken [Default = on]

`\addrfield` schaltet Adreß- und „Location“-Feld [Default = on]

`\subject` schaltet „Betreff“-Text [Default = off]

6 Das scrpage-package

Das scrpage-package definiert eine neue Schnittstelle um das Seitenlayout wesentlich flexibler definieren zu können, als dies mit dem Befehl `\pagestyle` möglich ist. Es ähnelt in dieser Hinsicht dem fancyheadings-package [POost94], beschreitet aber in vielerlei Hinsicht neue Wege. So existiert z. B. neben der normalen, einfachen Anwenderschnittstelle auch eine komplexere, aber auch flexiblere für die Experten unter uns.

Das scrpage-package funktioniert hierbei nicht nur mit den KOMA-classes, sondern auch mit den LaTeX-classes article, report und book.

6.1 Die Anwenderschnittstelle

`\headfont`
`\pnumfont`

Der Befehl `\headfont` wird benutzt, um auf den für die Seitenköpfe und -füße definierten Zeichensatz umzuschalten; `\pnumfont` hingegen bestimmt den Zeichensatz der Seitennummern. Beide Befehle sind mit `\normalfont` vorbelegt, können aber natürlich auf die eigenen Bedürfnisse umdefiniert werden, beispielsweise mit

```
\renewcommand{\headfont}{\normalfont\slshape}   .
```

`\headmark`

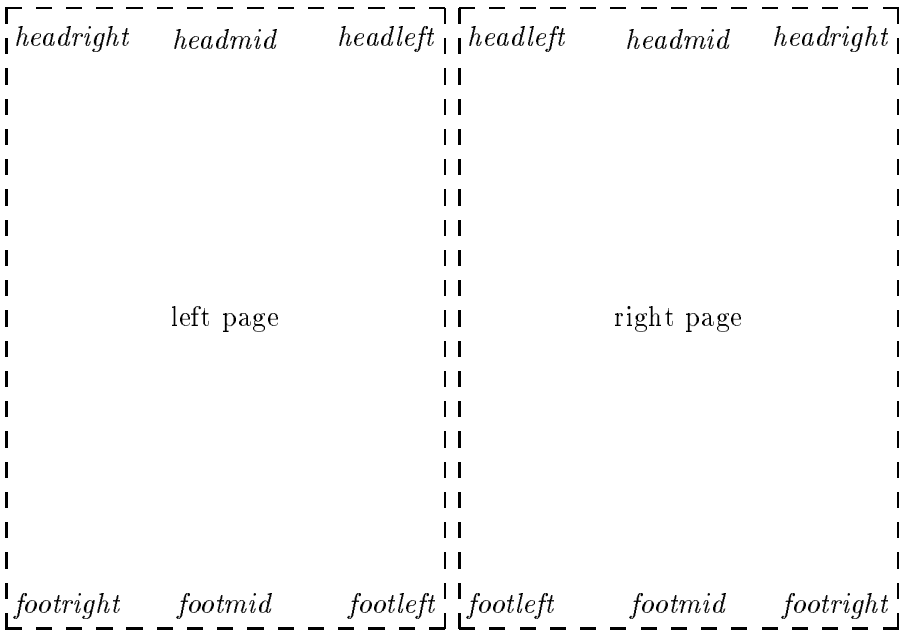
Wird mittels der Option `twoside` ein zweiseitiges Layout gewählt, so entspricht der Befehl `\headmark` dem LaTeX-Befehl `\leftmark` auf geraden bzw. `\rightmark` auf ungeraden Seiten. Bei einseitigem Layout entspricht `\headmark` immer `\rightmark`. Das scrpage-package nimmt einem also die Entscheidung, ob es sich bei der aktuellen Seite um eine gerade oder ungerade Seite handelt, ab, so daß `\headmark` in jedem Falle die aktuelle Kopfzeile enthält.

`\pagemark`

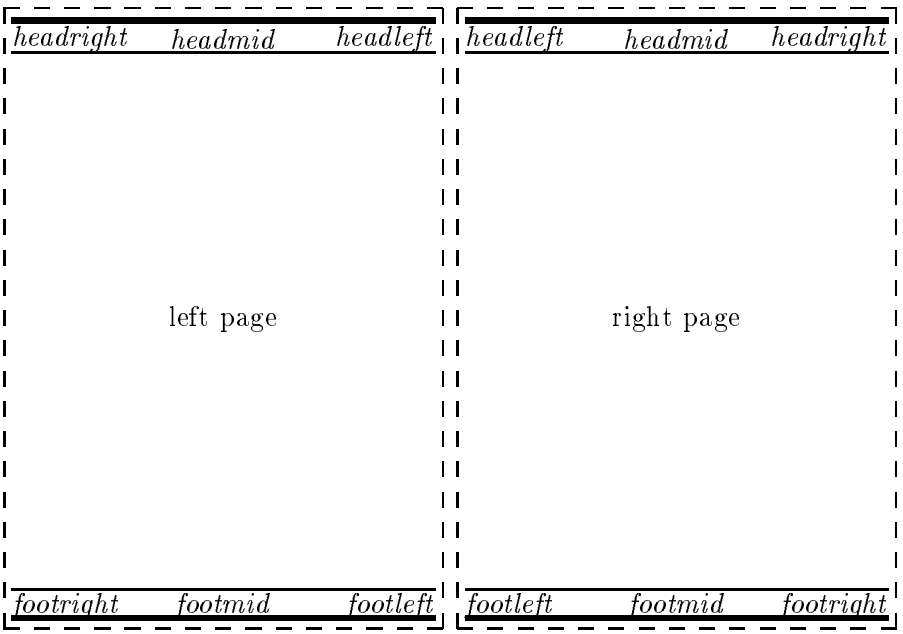
`\pagemark` enthält immer die aktuelle Seitenzahl. Im Gegensatz zu `\thepage` wird bei `\pagemark` automatisch der richtige Zeichensatz (mittels `\pnumfont`) ausgewählt.

`\deftripstyle{`*Name*`}[`*olw*`][`*ilw*`]{`*headleft*`}{`*headmid*`}{`*headright*`}{`*footleft*`}{`*footmid*`}{`*footright*`}`

Mittels `\deftripstyle` wird nun ein neues Seitenlayout namens *Name* definiert. Die Argumente *headleft*, *headmid*, *headright*, *footleft*, *footmid* und *footright* verteilen sich dabei wie folgt auf die Seiten:



Mittels der optionalen Argumente *ilw* (*i*nner *l*ine *w*idth, innere Linienbreite) und *olw* (*o*uter *l*ine *w*idth, äußere Linienbreite) können Separationslinien für den Kopf- und Fußteil der Seite erzeugt werden, hierfür ist die gewünschte Liniendicke anzugeben. Wird nur ein optionales Argument angegeben, so wird damit *ilw* gesetzt, also nur eine innere Separationslinie erzeugt (wie wir dies ja bereits von der Option `headsepline` der KOMA-classes kennen). Das ganze hat man sich dann so vorzustellen (mit $olw = 2\,\mathrm{pt}, ilw = 0{,}5\,\mathrm{pt}$):



Egal ob man mit `\deftripstyle` einen neuen Seitenstil definiert oder einen bereits existierenden umdefinitiert, in jedem Falle wird er erst bei Benutzung von `\pagestyle` oder `\thispagestyle` verwendet.

Zur Umdefinition des Seitenstils `myheadings` sollte allerdings folgender Befehl zur Verwendung kommen:

`\deftripstyle*{`*Name*`}[`*olw*`][`*ilw*`]{`*headleft*`}{`*headmid*`}{`*headright*`}{`*footleft*`}{`*footmid*`}{`*footright*`}`

Hiermit kann eine *my*-Variante des Seitenstils definiert werden. Diese zeichnet sich dadurch aus, daß der Seitenkopf nicht automatisch durch Kolumnentitel (wie `\chapter` und `\section`) verändert wird, sondern lediglich durch die Befehle `\markleft`, `\markright` und `\markboth` bestimmt wird.

6.2 Die Expertenschnittstelle

`\defpagestyle{`*Name*`}{`*Kopf-Definition*`}{`*Fuß-Definition*`}`
`\newpagestyle{`*Name*`}{`*Kopf-Definition*`}{`*Fuß-Definition*`}`
`\renewpagestyle{`*Name*`}{`*Kopf-Definition*`}{`*Fuß-Definition*`}`
`\providepagestyle{`*Name*`}{`*Kopf-Definition*`}{`*Fuß-Definition*`}`

All diese Kommandos definieren einen neuen Seitenstil, hierbei haben *Kopf-Definition* und *Fuß-Definition* folgenden Aufbau:

(*ull,ulw*){*gerade Seite*}{*ungerade Seite*}{*einseitig*}(*bll,blw*)

dabei sind uns *ulw* (*u*pper *l*ine *w*idth, obere Linienbreite) und *blw* (*b*elow *l*ine *w*idth, untere Linienbreite) bereits als *olw* und *ilw* begegnet, *ull* und *bll* sind die dazugehörigen Längen, die bei den Befehlen `\deftripstyle` und `\deftribstyle*` automatisch auf die Breite der aktuellen Seite (also `\textwidth`) gesetzt werden. Für *gerade Seite* und *ungerade Seite* sind die entsprechenden Definitionen der Seitenköpfe bzw. -füße einzusetzen. Ist das Seitenlayout einseitig, so wird statt diesen Definitionen immer *einseitig* verwendet. Was bisher nicht erläutert wurde sind die Unterscheidungsmerkmale der einzelnen Befehle, dies soll jetzt nachgeholt werden:

`\defpagestyle` definiert *immer* den Seitenstil *Name*, egal ob dieser vorher eventuell schon (anders) definiert war oder nicht. Dies stellt also quasi die Rambo-Variante dar, während `\newpagestyle` eine Fehlermeldung ausgibt, falls der Seitenstil schon definiert sein sollte. Mit `\renewpagestyle` verhält es sich genau andersherum: Ist der Seitenstil noch nicht definiert, ergibt sich hier eine Fehlermeldung, ansonsten wird er wie angegeben umdefiniert. `\providepagestyle` zu guter letzt definiert den angegebenen Seitenstil nur, wenn er vorher noch nicht definiert war. Im Gegensatz zu `\newpagestyle` wird aber bei bereits vorhandener Definition nix getan (stimmt nicht ganz, es wird immerhin ein Eintrag in der LOG-Datei getätigt), also insbesondere keine Fehlermeldung ausgegeben.

Den alten LaTeX 2ε-Hasen unter uns wird dies alles bekannt vorkommen, sind diese Befehle doch analog zu `\def`, `\newcommand`, `\renewcommand` und `\providecommand` ausgelegt worden.

So, nach so viel trockener (aber hoffentlich schmeckender) Theorie nun endlich, quasi als krönender Abschluß, ein Beispiel, mit dem die Kopf- und Fußzeilen dieses Kapitels realisiert wurden:

```
\renewpagestyle{plain}{(\textwidth,0pt)%
                        {\hfill}{\hfill}{\hfill}%
                        (\textwidth,0pt)}%
                       {(\textwidth,.4pt)%
                        {\pagemark\hfill}%
                        {\hfill\pagemark}%
                        {\hfill\pagemark\hfill}%
                        (\textwidth,1pt)}
\renewpagestyle{headings}{(\textwidth,1pt)%
                 {\headmark\hfill}{\hfill\headmark}{\hfill\headmark\hfill}%
                 (\textwidth,.4pt)}%
                {(\textwidth,.4pt)%
                 {\pagemark\hfill%
                  Copyright \copyright\ Markus Kohm, 1994--1997}%
                 {Package \texttt{scrpage}\hfill\pagemark}%
                 {\rlap{Package \texttt{scrpage}}\hfill%
                  Copyright \copyright\ Markus Kohm, 1994--1997\hfill%
                  \llap\pagemark}%
```

```
                (\textwidth,1pt)}
\pagestyle{headings} % und ab geht die Post...
```

6.3 Eigene Seitenstile und Kapitelanfangsseiten

Bei der Definition eigener Seitenstile gibt es einen weiteren Punkt zu beachten. Alle mir bekannten Klassen schalten bei Kapitelanfangsseiten automatisch mit Hilfe von `\thispagestyle{plain}` für diese eine Seite auf den Seitenstil `plain` um. Dies betrifft außerdem auch weitere Befehle, die einen Kapitelanfang beinhalten (beispielsweise `\tableofcontens`). Definiert man nun also einen eigenen Seitenstil und schaltet diesen ein, so wird auf allen Kapitelanfangsseiten trotzdem noch ein anderer, möglicherweise absolut unpassender, Seitenstil verwendet. Um dies zu ändern, muß der Seitenstil `plain` ebenfalls umdefiniert werden, wie dies im Beispiel im vorhergehenden Abschnitt vorgeführt wurde.

Bei der Umdefinierung von `plain` stellt sich nun die Frage, ob hier ein normaler oder eine *my*-Variante definiert werden soll. Dies richtet sich ganz einfach nach dem verwendeten Hauptseitenstil.

7 Die kleineren Beigaben

Dem KOMA-Script Paket liegen noch ein paar kleine, aber feine packages bei, die dem Anwender Freude bereiten und (ebenso wie das scrpage-package) auch in Verbindung mit den original LaTeX-classes problemlos funktionieren.

7.1 Das scrdate-package

`\todaysname`
`\dayname{`*Name*`}`

Mit `\todaysname` erhält man neben dem Tag, den man ja mit `\today` erhält und mit `\date` bekanntermaßen setzen kann, auch den dazugehörigen Wochentag. So liefert beispielsweise:

```
Dieses Dokument wurde an einem \todaysname\ "ubersetzt.
```

die Ausgabe

Dieses Dokument wurde an einem Dienstag übersetzt.

`\dayname` setzt entsprechend diesen Wochentag auf einen festen Wert.

Das scrdate-package beherrscht derzeit die Sprachen Englisch (english and USenglish), Deutsch (german) und Französisch (french), Italienisch (italian) und Spanisch (spanish), kann aber auch für andere Sprachen konfiguriert werden. Näheres dazu entnehme man `scrdate.dtx`.

Bei der aktuellen Version ist es egal, ob scrdate vor oder nach german, babel oder ähnlichen Paketen geladen wird, in jedem Falle wird die korrekte Sprache gewählt.[1]

[1] Etwas genauer ausgedrückt: Solange die Spachauswahl in einer zu babel bzw. german kompatiblen Form erfolgt und die Sprache scrdate bekannt ist, wird die Sprache korrekt gewählt. Ist dies nicht der Fall, werden (US)englische Ausdrücke verwendet. Wie man scrdate neue Sprachen beibringen kann, ist `scrdate.dtx` zu entnehmen.

7.2 Das scrtime-package

`\thistime[`*Trennung*`]`
`\thistime*[`*Trennung*`]`
`\settime{`*Wert*`}`

`\thistime` liefert die aktuelle Zeit; als Trennbuchstabe zwischen den Werten Stunden, Minuten und Sekunden wird das optionale Argument *Trennung* verwendet; die Voreinstellung ist hierbei das Zeichen ':'.

`\thistime*` funktioniert fast genau wie `\thistime`. Der einzige Unterschied besteht darin, daß im Gegensatz zu `\thistime` bei `\thistime*` die Minutenangaben bei Werten kleiner 10 nicht durch eine vorangestellte Null auf zwei Stellen erweitert wird.

So liefert beispielsweise die Zeile

```
Ihr Zug geht um \thistime Uhr.
```

als mögliche Ergebnisse:

Ihr Zug geht um 11:20Uhr.

Ihr Zug geht um 23:09 Uhr.

Demgegenüber liefert die Zeile

```
Beim n"achsten Ton ist es \thistime*[\ Uhr,\ ] Minuten und 42 Sekunden.
```

als mögliche Ergebnisse:

Beim nächsten Ton ist es 11 Uhr, 20 Minuten und 42 Sekunden.

Beim nächsten Ton ist es 23 Uhr, 9 Minuten und 42 Sekunden.

`\settime` setzt diese Zeit auf einen bestimmten festen Wert[2]; anschließend wird das optionale Argument von `\thistime` bzw. `\thistime*` ignoriert, da ja die Zeichenkette, die `\thistime` bzw. `\thistime*` nun liefert, hiermit explizit festgelegt wurde.

`12h`
`24h`

Mit den Optionen `12h` und `24h` kann ausgewählt werden, ob die Zeit bei `\thistime` und `\thistime*` im 12-Stunden- oder 24-Stunden-Format ausgegeben werden soll; voreingestellt ist `24h`.[3] Die Option verliert bei einem Aufruf von `\settime` ebenfalls ihre Gültigkeit.

[2] Allerdings darf man nicht erwarten, daß nun die Zeit stillsteht!

[3] Leider beherrscht das scrtime-package noch nicht die Sternzeit nach StarTrek, ein echter Mangel!

Literaturverzeichnis

[Duden20] DUDEN „Die deutsche Rechtschreibung“, Band 1, 20. Auflage, Dudenverlag, 1991

[CJK] WERNER LEMBERG: „Chinesisch, Japanisch und Koreanisch mit LaTeX“, 1995

[POost94] PIET VAN OOSTRUM: „Page headers and footers in LaTeX“, 11. Oktober 1994

[JTsch87] JAN TSCHICHOLD: „Ausgewählte Aufsätze über die Gestalt des Buches und der Typographie“, Zweite Auflage, Birkhäuser Verlag, Basel 1987

Index

Symbols

BibTeX 41
12h (Option) *54*
24h (Option) *54*

A

a0paper (Option) *13–14*
Abbildungen 20, 23, 24
abstractoff (Option) *21*
abstracton (Option) *21*
\addchap *25*
\addchap* *25*
addrconv.bst 41
addrconv.tex 42
\address 45
\Address *41*
\addrfield 47
\addrfieldoff *34*
\addrfieldon *34*
\addrheight 47
\addrindent 47
\addrvskip 47
\addrwidth 47
\addsec *25*
\addsec* *25*
\adrentry *36–40*, 40
adrguide.tex 41
\areaset *12*, 13, 14
\AtEndOfClass 22

B

b0paper (Option) *13–14*
babel (package) 53
Babel (package) 31
\backaddress *34*, 46
\baselineskip 8
BCOR 9, 10
BCOR (Option) *10–12*, 13
\beginletter 44
\bib@heading 23
bibtotoc (Option) *23*
bibtotocnumbered (Option) *23*
bigheadings (Option) *16*
birthday.bst 42
birthday.tex 42
Briefe 15, 16, 17, 28

C

c0paper (Option) *13–14*
\capfont *24*
\caplabelfont *24*
\caption 20, 24
\captionsamerican *30*
\captionsaustrian *30*, 44
\captionsbritish *30*
\captionsenglish *30*, 44
\captionsfrench *30*, 44
\captionsgerman *30*, 44
\captionsitalian *30*, 44
\captionsspahish *30*
\captionsspanish *30*
\captionsUKenglish *30*
\captionsUSenglish *30*, 44
\cc 29, 44
\ccname 45
CJK 27
\closing 29, 44
\Comment *41*
\customer *35*, 46
\customername 45

D

d0paper (Option) *13–14*
Danksagung 25
\date 53
\dateamerican *30*
\dateaustrian *30*, 44
\datebritish *30*
\dateenglish *30*, 44
\datefrench *30*, 45

\dategerman *30*, 44
\dateitalian *30*, 45
\datename 45
\datespanish *30*
\dateUKenglish *30*
\dateUSenglish *30*, 44
\dedication *21–22*
\deffootnote *20*
\deffootnotemark *20*
\defpagestyle *50–52*
\deftribstyle *49–50*
\deftribstyle* *50*
\descfont *24*
DIV 9, 10, 14
DIV (Option) *10–12*, 13
DIVcalc (Option) *10–12*
DIVclassic (Option) *10–12*
Duden 17

E

email.bst **42**
email.tex **42**
\encl 29, 44
\enclname 45
\endletter 44
enumerate (Umgebung) 20
\extratitle *21–22*

F

Faltmarken 47
Falzmarken 47
figure (Umgebung) 20
\firstfoot *36*, 46
\@firstfoot 46
\firsthead *36*, 46
\@firsthead 46
\FirstName *41*
\foldhskip 47
\foldmarks 47
\foldvskipi 47
\foldvskipii 47
\foldvskipiii 47
footexclude (Option) *13*
footinclude (Option) *13*
\footincludeoff 13, 16
\footincludeon 13, 16
footnosepline (Option) *18*
footsepline (Option) *18*
\FreeI *41*
\FreeII *41*
\frenchspacing 20
\fromaddress 45
\frombackaddress 46
\fromlocation 46
\fromname 45
\fromplace 46
\fromsig 45
Fußnoten 18, 19
Fußzeile 12, 17, 18

G

german (package) 20, 31, 53
Gliederung 16, 17, 25

H

headexclude (Option) *13*
\headfont *24*, *48*
headinclude (Option) *13*
\headincludeoff 13, 16
\headincludeon 13, 16
\headmark *48*
headnosepline (Option) *18*
headsepline (Option) *18*
\headtoname 45

I

\idx@heading 23
idxtotoc (Option) *23*
Index 23
Inhaltsverzeichnis 17, 22, 23, 52
\InputAddressFile *40*
\invoice *35*, 46
\invoicename 45
\isopaper *13–14*
itemize (Umgebung) 20

K

Kapitel 16, 52
Kolumnentitel 12, 17, 18, 24
Kompatibilitätsmodus 16
Kopfzeile 12, 17, 18

L

labeling (Umgebung) *26*

\LastName *41*
\listoffigures 23
\listoftables 23
liststotoc (Option) *23*
Literatur 23
\location *34*, 46
\locwidth 47
\lof@heading 23
\lot@heading 23
\lowertitleback *21–22*

M

\makefootnote *19*
\maketitle *21–22*
\marginline *24*
\marginpar 24
\minisec *25–26*
\myref *35*, 46
\myrefname 45

N

\name 29, 45
\Name *41*
\newpagesyle *50–52*
\nextfoot *36*, 46
\@nextfoot 46
\nexthead *36*, 46
\@nexthead 46
normalheadings (Option) *16*
Numerierung 16, 17, 23, 25, 27

O

\opening 29, 44
Option
  12h *54*
  24h *54*
  a0paper *13–14*
  abstractoff *21*
  abstracton *21*
  b0paper *13–14*
  BCOR *10–12*, 13
  bibtotoc *23*
  bibtotocnumbered *23*
  bigheadings *16*
  c0paper *13–14*
  d0paper *13–14*
  DIV *10–12*, 13
  DIVcalc *10–12*
  DIVclassic *10–12*
  footexclude *13*
  footinclude *13*
  footnosepline *18*
  footsepline *18*
  headexclude *13*
  headinclude *13*
  headnosepline *18*
  headsepline *18*
  idxtotoc *23*
  liststotoc *23*
  normalheadings *16*
  orgdate *31*
  pointednumbers *17*
  pointlessnumbers *17*
  slocfield *34*
  smallheadings *16*
  twoside *32*
  wlocfield *34*
orgdate (Option) *31*

P

package
  babel 53
  Babel 31
  german 20, 31, 53
  scraddr *40–41*
  scrdate *53*
  scrpage *48–52*
  scrtime *54*
  typearea 9, 13, 14, 15
\pagemark *48*
\pagename 45
Pagina 17, 24
\paperheight 14
\paperwidth 14
Papierformat 11, 13, 14
\place 35, 46
\pnumfont *24*, *48*
pointednumbers (Option) *17*
pointlessnumbers (Option) *17*
\providepagestyle *50–52*
\ps 29, 44
\publishers *21–22*

R

\raggedright 17
\raggedsection *17*
Randnotizen 24
\refitemi *35*, 46
\refitemii *35*, 46
\refitemiii *35*, 46
\refitemnamei *35*, 46
\refitemnameii *35*, 46
\refitemnameiii *35*, 46
\refvskip 47
\renewpagestyle *50–52*

S

Satzspiegel 9, 12, 15
  automatisch 10, 11
  Briefe 32
  klassisch 11
  manuell 12
  voreingestellt 10, 14
Schmutztitel 21
Schriftart 24
Schriftgröße 16
scraddr (package) *40–41*
scrdate (package) *53*
Script 2.0 ..... 5, 6, 13, 15, 17, 18
scrlettr 16
scrpage (package) *48–52*
scrtime (package) *54*
\sectfont *24*
Seitenstil 17, 36, 52
\selectlanguage 31
\SetDIVList *14–15*
\settime *54*
\sfb *7*
\sigindent 47
\signature 29, 45
slocfield (Option) *34*
smallheadings (Option) *16*
\specialmail *34*, 46
\@specialmail 46
Speicher 22
StarTrek 54
\subject *21–22*, *35*, 46, 47
\@subject 46
\subjectname 45
\subjectoff *35*
\subjecton *35*

T

Tabellen 20, 23, 24
table (Umgebung) 20
\telephone 46
\Telephone *41*
\telephonenum 46
Textbereich 12
\textheight 8
\textsuperscript *19–20*
\textwidth 8
\thefootnotemark *19–20*
\thistime *54*
\thistime* *54*
Titel 20, 21
Titelrückseite 22
\title *35*, 46
\@title 46
\titlehead *21–22*
\today 53
\todaysname *53*
\topskip 8
twoside (Option) *32*
\typearea *9–12*, 12, 13, 14, 15
typearea (package) 9, 13, 14, 15
typearea.cfg 14, 15

U

Überschriften 16, 17, 18, 20, 24
Umgebung
  enumerate 20
  figure 20
  itemize 20
  labeling *26*
  table 20
Unterschriften 20, 24
\uppertitleback *21–22*

V

Vorwort 25

W

Widmung 22
wlocfield (Option) *34*

Y

`\yourmail` *35*, 46
`\yourmailname` 45
`\yourref` *35*, 46
`\yourrefname` 45

Z

Zeilenlänge 12
Zusammenfassung 21